

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-281065
(P2002-281065A)

(43) 公開日 平成14年9月27日 (2002.9.27)

(51) Int.Cl.⁷

H 0 4 L 12/56

識別記号

F I

H 0 4 L 12/56

テーマコード* (参考)

Z 5 K 0 3 0

審査請求 未請求 請求項の数32 書面 外国語出願 (全 59 頁)

(21) 出願番号 特願2001-197685(P2001-197685)

(22) 出願日 平成13年6月29日 (2001.6.29)

(31) 優先権主張番号 09/607807

(32) 優先日 平成12年6月30日 (2000.6.30)

(33) 優先権主張国 米国 (U S)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 山本 彰

アメリカ国カリフォルニア州クパチーノ
ルッペルプレイス1202

(72) 発明者 山神 憲司

アメリカ国カリフォルニア州ロスガトス
カイルニベル108

(74) 代理人 100075096

弁理士 作田 康夫

Fターム(参考) 5K030 GA11 HA08 HC01 HC14 JT06
KA02

(54) 【発明の名称】 データの完全性を伴いデータネットワークに接続される記憶装置システム

(57) 【要約】

【解決手段】 通信ネットワークによってリモート記憶装置コンポーネントに結合されたローカル記憶装置コンポーネントを備える記憶装置システム。通信ネットワークは、データパケットが送信されたものと同じ順序でのデータパケットの受信を保証できないことにより特徴づけられる。

【効果】 本発明に従った方法は、当該通信ネットワークの性質にもかかわらず、受信された要求を処理する適切な順序づけを保証する。

【図1】

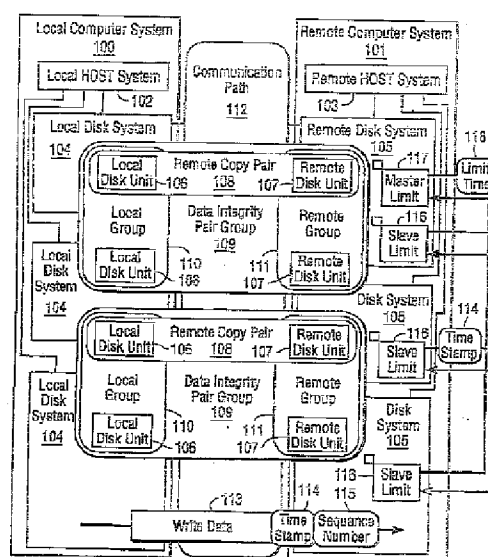


FIG. 1

【特許請求の範囲】

【請求項1】 記憶装置システムであって、
第1の記憶装置コンポーネントを有する第1のコンピュータシステムと、
第2の記憶装置コンポーネントを有する第2のコンピュータシステムとを含み、
第1および第2の記憶装置コンポーネントはデータネットワークによってデータを交換するように構成されており、
第1のコンピュータシステムは、データのブロックを第1の記憶装置コンポーネントに書き込み、データパケットを第2のコンピュータシステムに伝送するためにプログラムコードで構成されたメモリを有しており、データパケットはデータブロック、タイムスタンプおよびシーケンス番号を含むものであり、
第2のコンピュータシステムは、第1のコンピュータシステムからデータパケットを受信し、データパケットに含まれているタイムスタンプおよびシーケンス番号に基づき候補データパケットを選択し、候補データパケットを第2の記憶装置システムに書き込むようにプログラムコードにより構成されたメモリを有しており、
ここにおいて、第1の記憶装置コンポーネントに書込まれるデータのブロックは、第1の記憶装置コンポーネントにおける同じ順序で第2の記憶装置コンポーネントに書込まれるものである、システム。

【請求項2】 第2のメモリがさらに、タイムスタンプの中からそれらの対応するシーケンス番号に基づきリミットタイムスタンプを取得し、それらの対応するタイムスタンプをリミットタイムスタンプと比較することによってデータパケットの中から候補データパケットを選択するようにプログラムコードにより構成されている、請求項1記載のシステム。

【請求項3】 データネットワークがコネクションレス型ネットワークである、請求項1記載のシステム。

【請求項4】 データパケットがそれらが送信されたものと同じ順序で受信されると保証できないものとしてデータネットワークが特徴づけられる、請求項1記載のシステム。

【請求項5】 データネットワークがワイドエリアネットワークである、請求項4記載のシステム。

【請求項6】 第1の記憶装置コンポーネントが複数の第1のデータ記憶装置よりなり、第2の記憶装置コンポーネントが複数の第2のデータ記憶装置よりなり、第1のデータ記憶装置の各々は第2のデータ記憶装置の1個と対応しており、ここにおいて、第1のデータ記憶装置の1個に格納されたデータが対応する第2のデータ記憶装置の1個にも格納される、請求項1記載のシステム。

【請求項7】 第1の記憶装置コンポーネントが複数の第1のディスクシステムよりなり、第2の記憶装置コンポーネントが複数の第2のディスクシステムよりなり、

第1のディスクシステムは各々第2のディスクシステムの1個以上と関係づけられており、ここにおいて、第1のディスクシステムのうちの1個に格納されたデータが関係する1個以上の第2のディスクシステムにも格納される、請求項1記載のシステム。

【請求項8】 第1のディスクシステムの各々が複数の第1のディスク装置よりなり、第2のディスクシステムの各々が複数の第2のディスク装置よりなり、第1のディスク装置の各々は第2のディスク装置の1個と関係づけられている、請求項7記載のシステム。

【請求項9】 第1のディスク装置は各々、第1のディスク装置が属する第1のディスクシステムとは独立に、第2のディスク装置のうちの1個と関係づけられている、請求項8記載のシステム。

【請求項10】 ローカルシステムに含まれるデータをリモートシステムにバックアップする方法であって、データのブロックをローカルデータ記憶装置に書き込むことと、

データパケットをリモートシステムに送信することと、
データパケットはデータのブロック、タイムスタンプおよびシーケンス番号を含むものであり、
ローカルシステムからデータパケットを受信することと、

そのデータのブロックがリモートデータ記憶装置に書込まれるデータパケットを、データパケットのシーケンス番号およびタイムスタンプに基づき選択することを含む、方法。

【請求項11】 次のデータパケットのシーケンス番号を増分することをさらに含む、請求項10記載の方法。

【請求項12】 データパケットを選択することが、タイムスタンプの中からそれらの関係するシーケンス番号に基づきリミットタイムスタンプを取得することと、それらの関係するタイムスタンプをリミットタイムスタンプと比較することによってデータパケットの中からそのデータパケットを選択することを含む、請求項10記載の方法。

【請求項13】 ローカルデータ記憶装置が複数のローカルディスク装置よりなり、リモートデータ記憶装置が複数のリモートディスク装置よりなり、各ローカルディスク装置がリモートコピーペアを規定するためにリモートディスク装置の1個と対にされている、請求項10記載の方法。

【請求項14】 ローカルディスク装置にデータの複数のブロックを書き込むことと、各リモートディスク装置がその関係する複数のデータパケットに基づくシーケンス番号のリストを有するようにリモートディスク装置に複数のデータパケットを送信することとをさらに含み、該方法はさらに、シーケンス番号の各リストについて、シーケンス番号の最も長い連続を取得することと、その最も長い連続から最高値のシーケンス番号を取得すること

と、最高値のシーケンス番号に対応するタイムスタンプを取得することと、それによって、タイムスタンプのリストを生成することを含み、該方法はさらに、タイムスタンプのリストの中で最早タイムスタンプに基づきデータパケットを選択することを含む、請求項 1 3 記載の方法。

【請求項 1 5】 ローカルデータ記憶装置が複数のローカルディスクシステムよりなり、リモートデータ記憶装置が複数のリモートディスクシステムよりなり、各ローカルディスクシステムが 1 個以上のリモートディスクシステムと関係づけられており、ここにおいて、ローカルディスクシステムの 1 個以上に格納されたデータが、関係する 1 個以上のリモートディスクシステムにも格納される、請求項 1 0 記載の方法。

【請求項 1 6】 ローカルディスクシステムの各々が複数のローカルディスク装置よりなり、リモートディスクシステムの各々が複数のリモートディスク装置よりなり、ローカルディスク装置の各々がリモートディスク装置の 1 個と関係づけられている、請求項 1 5 記載の方法。

【請求項 1 7】 各ローカルディスク装置が、ローカルディスク装置が属するローカルディスクシステムとは独立に、リモートディスク装置の 1 個と関係づけられている、請求項 1 6 記載の方法。

【請求項 1 8】 データのブロックをローカルデータ記憶装置に書込むこととデータパケットをリモートシステムに送信することが非同期に実行される、請求項 1 0 記載の方法。

【請求項 1 9】 データパケットがコネクションレス型データネットワークによって送信される、請求項 1 0 記載の方法。

【請求項 2 0】 データパケットが、それらが送信されたものと同じ順序で行先に着信すると保証できないものとして特徴づけられるデータネットワークによってデータパケットが送信される、請求項 1 0 記載の方法。

【請求項 2 1】 データネットワークがワイドエリアネットワークである、請求項 2 0 記載の方法。

【請求項 2 2】 複数のローカルデータ記憶装置よりなるローカル記憶装置システムにおいて、ローカル記憶装置システムのデータを、複数のリモートデータ記憶装置よりなるリモート記憶装置システムにバックアップする方法であって、該方法が、各ローカルデータ記憶装置において、それに書込まれるデータブロックを受信することと、各ローカルデータ記憶装置において、データブロック、タイムスタンプおよびシーケンス番号よりなるデータパケットをリモートデータ記憶装置の 1 個に伝送することと、リモートデータ記憶装置において、ローカルデータ記憶装置から複数のデータパケットを受信することと、ここ

において、各リモートデータ記憶装置は、その関係する複数のデータパケット、シーケンス番号のリストおよび、関係するデータパケットからのタイムスタンプのリストを有するものであり、

各リモートデータ記憶装置において、シーケンス番号の中で最も長い連続を識別し、最も長い連続の最高値のシーケンス番号のデータパケットを取得することと、各リモートデータ記憶装置において、得られたデータパケットから最早タイムスタンプを取得することと、得られたタイムスタンプのうちで最早のものをリミットタイムとして選択することと、各リモートデータ記憶装置において、最早タイムスタンプを有する候補データパケットを選択することと、タイムスタンプがリミットタイムより早い候補データパケットの中からデータパケットを選択することを含む、方法。

【請求項 2 3】 ローカル記憶装置システムが 1 個以上のローカルディスクシステムよりなり、各ローカルディスクシステムが 1 個以上のローカルディスク駆動装置よりなり、リモート記憶装置システムが 1 個以上のリモートディスクシステムよりなり、各リモートディスクシステムが 1 個以上のリモートディスク駆動装置よりなり、各ローカルディスク駆動装置がリモートコピーペアを規定するためにリモートディスク駆動装置の 1 個と関係づけられている、請求項 2 2 記載の方法。

【請求項 2 4】 各ローカルデータ記憶装置がローカルディスク駆動装置の 1 個であり、各リモートデータ記憶装置がリモートディスク駆動装置のうちの 1 個であり、ここにおいて、各リモートコピーペアと関係づけられたシーケンス番号が存在する、請求項 2 3 記載の方法。

【請求項 2 5】 受信された複数のデータパケットがリモートコピーペアに従ってグループ化される、請求項 2 4 記載の方法。

【請求項 2 6】 シーケンス番号が、共通のリモートコピーペアを有するローカルおよびリモートディスクシステムの各対と関係づけられている、請求項 2 3 記載の方法。

【請求項 2 7】 複数のデータパケットの各々が、それが送信された元のローカルディスクシステムに基づきグループ化される、請求項 2 6 記載の方法。

【請求項 2 8】 各リモートコピーペアが複数のデータの完全性ペアの 1 個と関係づけられており、ここにおいて、シーケンス番号が、共通のデータの完全性ペアグループを有するローカルおよびリモートディスクシステムの各対と関係づけられている、請求項 2 3 記載の方法。

【請求項 2 9】 データアクセス方法であって、データ転送要求を付与することと、データ転送要求はシーケンス番号を含むものであり、データ転送要求をローカルシステムからリモートシステムに伝送することと、

リモートシステムにおいて、データのブロックを含んでいるエントリの待ち行列を付与することと、データ転送要求は待ち行列のターゲットエントリに向けられており、

リモートシステムにおいて、データ転送要求のシーケンス番号をシーケンス番号カウンタの現在値と比較することと、

シーケンス番号がシーケンス番号カウンタと等しくない場合、シーケンス番号とシーケンス番号カウンタとの差に基づくエントリの数だけ待ち行列をトラバースすることによって待ち行列のターゲットエントリへのアクセスを得ることと、

シーケンス番号がシーケンス番号カウンタに等しい場合、ターゲットエントリへのアクセスを得るために待ち行列の一方の端にアクセスすることと、

ターゲットエントリでデータ転送要求を実行することとを含む、方法。

【請求項 30】 リモートシステムが磁気テープ記憶装置システムを含み、データ転送要求が磁気テープ記憶装置システムへの読出しおよび書込み要求である、請求項 29 記載の方法。

【請求項 31】 データ転送要求が読出し要求であり、待ち行列のデータのブロックが読出し要求を満たすために使用される、請求項 29 記載の方法。

【請求項 32】 データ転送要求が、新しいエントリとして待ち行列に挿入される書込みデータを含む書込み要求であり、新しいエントリがターゲットエントリの前または後に挿入される、請求項 29 記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ記憶装置システム、詳細には、データネットワーク環境においてデータ記憶装置システムのデータの完全性を維持することに関する。

【0002】

【従来の技術】従来、データ処理システムは、高速な高信頼性データバスによって各自の関係するデータ記憶装置システムにアクセスできる。しかし、その機会は、ネットワーク通信の広範囲に及ぶ用途が拡大し続けるにつれ、ますます利用可能になっている。IP（インターネットプロトコル）は、TCP/IP（伝送制御プロトコル/インターネットプロトコル）ネットワークが確立される基本的なパケットデリバリーサービスを提供する。IPは、十分に規定されたプロトコルであり、それゆえ、IPを用いてサーバシステムが記憶装置システムとデータを交換し、記憶装置システムが他の記憶装置システムとデータを交換する、ネットワークベースのデータ記憶アクセスのためのトランスポート/ネットワーク層を提供する自然な候補である。

【0003】しかし、IPの性質は、データ記憶アクセ

スシステムの領域におけるいくつかの独自の問題を呈する。第一に、IPはコネクションレス型プロトコルである。これは、IPがデータの伝送に先んじてエンドツーエンド接続を確立するために制御情報を交換しないことを意味する。IPは誤り検出・回復機構をまったく含まない。従って、IPは接続されたネットワークにデータを送達するためには信頼できるが、そのデータが正しく受信されたか、またはそれが送信された順序でデータが受信されているかを保証するためのいかなる機構も存在しない。コネクション型サービスが要求される場合、IPは、接続を確立するためにより高次の階層プロトコルに頼る。

【0004】デュアルリモートコピー機能が必要とされるデータ記憶装置システムにおいて、IPベースの伝送は問題を呈する。リモートコピー機能は、一次データ記憶装置での災害回復を実現するという目的で、リモートサイトにおける一次データ記憶装置のリアルタイムコピーを可能にする。この機能がその目的を果たすために、データの完全性を保証することが重要である。同期および非同期の2種類のリモートコピーがある。

【0005】同期型リモートコピーでは、ローカルHOSTによるその関係するローカルディスクシステムへの書込み要求は、書込まれたデータがローカルディスクシステムからリモートディスクシステムに転送された後まで完了しない。従って、同期型コピーの場合、ローカルおよびリモートディスクシステム間のデータの完全性を保証することは容易である。

【0006】非同期型リモートコピーでは、ローカルHOSTによる書込み要求は、ローカルディスクがリモートディスクへのその転送を完了する前に、完了する。名称が暗示する通り、ローカルディスクからリモートディスクへの転送動作が完了しているかどうかにかかわらず、制御はローカルHOSTに返される。従って、非同期型コピー動作の間のデータの完全性は、リモートディスクのデータがローカルディスクにおける同じ順序で書込まれるように、リモートディスクシステムでのデータの正しい着信順序に依存する。

【0007】これを実現するために、ローカルディスクシステムは、リモートディスクに送られるデータとともにタイムスタンプを含める。リモートにおいてデータはそのタイムスタンプに従って書込まれる。従って例えば、リモートディスクがタイムスタンプ7:00を備えるデータを受信した時には、それはすでに、タイムスタンプが7:00より前である全部のデータを受信している。

【0008】

【発明が解決しようとする課題】しかし、IPベースのネットワークにおいて、パケットが乱順で着信できる場合、7:00のタイムスタンプを有するデータパケットは、それ以前のタイムスタンプを有するデータパケット

が先行することもあるかもしれない。その結果、伝送プロトコルがIPのようなコネクションレス型トランスポートモデルに基づく場合、リモートディスクシステムにおいてデータの完全性を保証することは難しい。

【0009】IPが磁気テープシステムで使用された時には別の問題が生じる。磁気テープへの読出しおよび書き込み動作はシーケンシャルであり、従って、アドレス指定は固定である。従ってデータパケットが乱順でリモートサイトに着信した場合、データは間違った順序でテープに書込まれる。破壊された記憶装置システムをテープから復元するための以降の回復動作は結果として破損データを生じるであろう。

【0010】信頼できるIPベースのデータ回復システムを提供するための必要性が存在する。

【0011】

【課題を解決するための手段】本発明に従ったデータ記憶装置システムは、ローカル記憶装置コンポーネントおよびリモート記憶装置コンポーネントよりなる。ローカル記憶装置コンポーネントに書込まれるデータは、データネットワークによってリモート記憶装置コンポーネントにデータパケットで送られる。

【0012】データパケットは、ローカル記憶装置コンポーネントで書込まれるデータのコピー、タイムスタンプおよびシーケンス番号を含む。複数のそうしたデータパケットがリモート記憶装置コンポーネントで受信される。データパケットは、各データパケットに関するシーケンス番号およびタイムスタンプに基づきリモート記憶装置コンポーネントでの書き込みのために選択される。

【0013】本発明の1実施形態において、ローカルおよびリモート記憶装置コンポーネントはそれぞれ、複数のローカルおよびリモートディスク装置として構成される。各ローカルディスク装置はリモートディスク装置と関係づけられる。そのような各対はリモートコピーペアと呼ばれる。この実施形態では、各リモートコピー単位は関係するシーケンス番号を有する。

【0014】本発明の別の実施形態において、ローカルディスク装置はローカルディスクシステムにグループ化される。同様に、各リモートディスク装置はリモートディスクシステムにグループ化される。本発明のこの実施形態では、少なくとも1個の共通のリモートコピーペアを有する各対のローカルおよびリモートディスクシステムについてシーケンス番号が存在する。

【0015】本発明の別の実施形態において、リモートコピーペアはデータの完全性ペアグループにグループ化される。この実施形態では、少なくとも1個のデータの完全性ペアグループを共通に有するローカルおよびリモートディスクシステムの各対についてシーケンス番号が存在する。

【0016】本発明の教示は、添付する詳細な説明を以

下の図面とともに検討することによって容易に理解することができる。

【0017】

【発明の実施の形態】図1を参照すれば、本発明の1実施形態に従ったコンピュータシステムが示されている。ローカルコンピュータシステム100は、少なくともローカルHOSTシステム102および少なくともローカルディスクシステム104を含む。リモートコンピュータシステム101は少なくともリモートディスクシステム105を含む。リモートHOSTシステム103は必ずしもリモートコンピュータシステム101に必要なわけではない。個々のローカルディスクシステム104は通信路112によってリモートディスクシステム105と接続される。通信路112は、伝送されたデータパケットが必ずしもそれらが送られた順序で着信するわけではないことを特徴とするネットワークである。IPベースのネットワークはこの振舞いを示す。一般に、コネクションレス型ネットワークはこの振舞いを示す。例えば、ワイドエリアネットワーク(WAN)はIPに基づくことができる。しかし、本発明はWANに限定されるものではない。

【0018】ローカルディスクシステム104は、リモートディスクシステム105において維持される関係するリアルタイムコピーを有する少なくともローカルディスク装置106を含む。リモートディスクシステム105は、ローカルディスク装置106のリアルタイムコピーを含んでいる少なくともリモートディスク装置107を含む。ローカルディスク装置106およびリモートディスク装置107よりなるペアは、リモートコピーペア108と称する。その中においてデータの完全性が保証されなければならないリモートコピーペア108のグループは、データの完全性ペアグループ109と称する。データの完全性ペアグループ109に属するローカルディスク装置106のグループは、データの完全性ローカルディスクグループ110と称する。同様に、リモートシステムでは、1個のデータの完全性ペアグループ109に属するリモートディスク装置107のグループは、データの完全性リモートディスクグループ111と称する。

【0019】データの完全性ローカルディスクグループ110は、単一のローカルディスクシステム104から、または、2個以上のローカルディスクシステム104からのローカルディスク装置106を含み得る。その結果、ローカルディスクシステム104の構成要素であるローカルディスク装置106は、1個以上のデータの完全性ペアグループ109に見ることができる。データの完全性リモートディスク装置グループ111は、1個以上のリモートディスクシステム105に属するリモートディスク装置107より構成され得る。従って、リモートディスクシステム105は、異なるデータの完全性

ペアグループ109に属するリモートディスク装置107を有することができる。

【0020】ローカルHOSTシステム102の動作の経過の間に、データはローカルディスクシステム104に書込まれる必要があるであろう。ローカルHOSTは、ローカルディスクシステム104に記憶される「書込みデータ」113を転送する。ローカルディスクシステム104はまた、書込みデータ113をデータ回復のためにリモートディスクシステム105にも送る。本発明によれば、ローカルディスクシステム104がリモートディスクシステム105に書込まれる書込みデータ113を送る際に、それはタイムスタンプ114およびシーケンス番号115も送る。タイムスタンプ114は、ローカルディスクシステムがローカルHOSTシステム102からその要求を受信した時刻を示す。シーケンス番号115は書込みデータ113のシーケンス番号である。シーケンス番号115は、ローカルディスク装置104が書込みデータ113をリモートディスクシステム105に送る時に生成される。ローカルディスクシステム104は、シーケンス番号114を選定し、それらがシーケンシャルに生成されていることを保証する。

【0021】データの完全性を実現するために、データの完全性ペアグループ109のリモートディスク装置107に書込まれるデータの順序は、そのデータの完全性ペアグループの対応するローカルディスク装置106に書込まれる同一データの順序と同じである必要がある。データの完全性ペアグループ109におけるデータの完全性を保証するために、リモートディスクシステム105の中でのタイムスタンプ114を比較することが必要である。なぜなら、リモートディスクシステム105のうちの1個が7:00のタイムスタンプ114を備える書込みデータ113を受信したのに対して、リモートディスクシステム105の別のものももっと早いタイムスタンプを備える書込みデータ113をすでに受信していることが起こり得るからである。

【0022】従って、リモートシステム101は、複数のスレーブリミットタイムスケジュールプロセス116を含む。各リモートディスクシステム105は、関係するスレーブリミットタイムスケジュールプロセス116を有する。リモートシステムはさらに、単一のマスタリミットタイムスケジュールプロセス117を含む。スレーブリミットタイムスケジュールプロセス116はそれぞれ、タイムスタンプ114の情報をマスタリミットタイムスケジュールプロセス117に送る。マスタリミットタイムスケジュールプロセス117はその後、リモートディスク装置107にデステージングを許可するために最も早い時間を決定する。この最早時間は、タイムリミット118として各スレーブタイムリミットスケジュールプロセス116に送られる。

【0023】本発明によれば、ローカルディスクシステ

ム104は、データパケットが乱順でリモートシステムに着信した場合でもリモートコピーシステムにおけるデータの完全性を保証するために、書込みデータ113の各パケットとともにシーケンス番号115を送る。

【0024】図2を参照すれば、本発明の実施形態において、この図は、すべてのリモートコピーペア108が1個のシーケンス番号を有する、ローカルディスクシステムおよびリモートディスクシステムにおけるプロセスを示す。ローカルディスクシステム104は、キャッシュメモリ202を有するローカルディスク制御装置200を備える。リモートディスクシステム105は、キャッシュメモリ203を備えるリモートディスク制御装置201を含む。

【0025】プロセスは、ローカルコンピュータシステム100において、L（ローカル）書込みデータ受信プロセス210、L書込みデータ送信プロセス211、L書込みデータ送信完了プロセス212およびL書込みデータデステージングプロセスを含む。リモートコンピュータシステム101は、R（リモート）書込みデータ受信プロセス214、R書込みデータデステージングプロセス215、マスタリミットタイムスケジュールプロセス117およびスレーブリミットタイムスケジュールプロセス116を含む。各プロセスは、その各自のローカルディスク制御装置200またはリモートディスク制御装置201において起動する。

【0026】各リモートコピーペア108について、ローカルディスクアドレス204、リモートディスクアドレス205、データの完全性グループID206およびシーケンスカウンタ207を含むリモートコピーペアデータ記憶装置203が存在する。ローカルディスクアドレス204およびリモートディスクアドレス205はそれぞれ、書込み動作のためにリモートコピーペア108を構成するローカルディスク装置およびリモートディスク装置の行先アドレスである。データの完全性ペアグループID206は、リモートコピーペアが属するデータの完全性ペアグループ109を識別する。図2に例示する通り、各リモートコピーペア108は関係するシーケンス番号115を有する。シーケンスカウンタ207は、リモートディスクに送られる書込みデータ113と関係づけられているシーケンス番号115を与える。

【0027】キャッシュメモリ202は、各リモートコピーペア108について、書込みデータ113、リモートディスクアドレス208、位置決め情報209、シーケンス番号115およびタイムスタンプ114よりなるデータ構造も含む。このように、ローカルHOST102からローカルディスクシステムに送られる書込みデータの各ブロックについて1個の当該データ構造が存在し、それによって、ローカルHOST102による典型的な書込み動作のために複数の当該データ構造を生じる。例えば、位置決め情報は、パーソナルコンピュータ

に共通に見られるディスク駆動装置のディスクブロックアドレスである。従来のメインフレームシステムでは、位置決め情報は一般に、シリンダ番号、ヘッド番号およびレコード番号である。

【0028】ここで図2および5を参照して、L書き込みデータ受信プロセス210の処理を説明する。このプロセスは、ローカルHOSTシステム102から書き込み要求を受信した時にローカルディスクシステムにおいて実行される。最初に、ローカルディスクシステム104はローカルHOSTシステム102から書き込み要求を受信する(ステップ500)。書き込み要求は、ローカルディスク装置106のアドレス情報および、書き込みデータ113が書込まれるローカルディスク装置106の位置を指定する。次に、ローカルディスクシステム104は、書込まれる実際の書き込みデータ113を受信し、それをキャッシュメモリ202にキャッシュする(ステップ501)。ローカルディスクシステム104は、リモートコピーペア108の対応するリモートディスク装置107のリモートディスクアドレス205をリモートコピーペアデータ記憶装置203から取得し、それをリモートディスクアドレス208としてキャッシュメモリに格納する。書き込み要求において指定された位置は、位置決め情報209としてキャッシュメモリに格納される(ステップ502)。ローカルディスクシステムはその後、ローカルHOSTシステムからタイムスタンプを受信し、それをタイムスタンプ114としてキャッシュメモリ202に格納する(ステップ503)。タイムスタンプ114は、生成されるタイムスタンプ114がローカルディスクシステム104の全部に共通である限り、ローカルHOSTシステム104以外によっても生成できる。複数のローカルHOSTシステム102が存在する実施形態では、HOSTシステム間で共通のタイムスタンプ値を付与する共有クロックが存在することが前提とされる。最後に、ローカルディスクシステム104は、ローカルHOSTシステム102に書き込み要求の完了を示す(ステップ504)。

【0029】図2および6を参照して、L書き込みデータ送信プロセス211の処理を説明する。このプロセスは、ローカルディスクシステムが書き込みデータ113をリモートディスクシステム105に送る準備ができた時に実行される。図2に示す本発明の実施形態によれば、各リモートコピーペア108について1個のL書き込みデータ送信プロセス211が存在する。このプロセスは、L書き込みデータ受信プロセス210に関して非同期に実行される。

【0030】ローカルディスクシステム104は、リモートディスクシステムに送られるリモートコピーペア108において待機している書き込みデータ113の全部のうちで時間的に最早関係するタイムスタンプ114を有する書き込みデータ113を選択する(ステップ60

0)。ローカルディスクシステムはその後、シーケンスカウンタ207の現在値を、選択された書き込みデータ113(すなわち、その関係するタイムスタンプが時間的に最も早い書き込みデータ)に関係づけられるシーケンス番号115とみなす。シーケンスカウンタ207は増分される(ステップ601)。次に、選択された書き込みデータ113ならびにその関係するタイムスタンプ114、シーケンス番号115、リモートディスクアドレス情報208および位置決め情報209は、リモートディスクシステム105に送られる(ステップ602、図1も参照)。

【0031】本発明によれば、L書き込みデータ送信プロセス211はその後、完了に関するリモートディスクシステム105からのいずれの指標も待たずに、次の書き込みデータ113を処理することに進む。このようにして、高速なデータ転送速度が実現される。しかし、パケットが乱順で着信する可能性がある。それゆえ、ローカルディスクシステム104は、リモートディスクシステム105に属するいずれかの書き込みデータ113が、まだ送られていないリモートコピーペア108に存在するかどうかを確認する(ステップ603)。肯定であれば、処理はステップ600に継続する。否定であれば、プロセスはしばらく待機し(ステップ604)、その後ステップ603に継続する。

【0032】ステップ604での待機は、送られる書き込みデータがまったく存在しない状況に適応するためである。ステップ603で送信する書き込みデータがまったく存在しなければ、L書き込みデータ送信プロセス211は行うべき処理がまったくない。従って、ステップ604は、送信すべき書き込みデータがあるかどうかを知るために再び確認する前にプロセスを暫時休止するために使用される。

【0033】次に図2および7を参照して、L書き込みデータ送信完了プロセス212の処理を説明する。このプロセスは、ローカルディスクシステム104がリモートディスクシステム105への書き込みデータ113の転送の完了の通知を受信した時に実行される。ローカルディスクシステムは、リモートディスクシステム105から書き込みデータ113の転送の通知を受信する(ステップ700)。ローカルディスクシステム104はその後、対応するタイムスタンプ114およびシーケンス番号115の値をヌルにする。書き込みデータ113はすでにリモートディスクシステム105に送られているので、これらのフィールドはもはやローカルディスクシステム104によって使用されない。

【0034】リモートディスクシステム104への書き込みデータ113の転送が完了した後、書き込みデータ113はL書き込みデータデステージプロセス213によってローカルディスク装置104に書込まれる。このプロセスは、ローカルディスク装置106への書き込みデータ1

13 の実際の書込みをもたらし、公知の常法に従って実行される。

【0035】次に図2および8を参照して、R書込みデータ受信プロセス214を検討する。このプロセスは、ローカルディスクシステム104から書込みデータ113を受信した時に、リモートディスクシステム105において実行される。図2に示す本発明の実施形態によれば、各リモートコピーペア108についてR書込みデータ受信プロセス214が存在する。リモートディスクシステム105は、受信された書込みデータ113ならびにその関係するタイムスタンプ114、シーケンス番号115、リモートディスクアドレス情報208および位置決め情報209をキャッシュメモリ203に格納する（ステップ800）。リモートディスクシステム105は、そのシーケンス番号115とともに書込みデータ113の転送の完了指標をローカルディスクシステムに返送する（ステップ801）。

【0036】キャッシュメモリ203は、各リモートコピーペア108について、受信された書込みデータ113、リモートディスクアドレス208、位置決め情報209、シーケンス番号115およびタイムスタンプ114よりなるデータ構造を含む。従って、ローカルHOST102から受信された書込みデータ113の各ブロックについて1個のそのようなデータ構造が存在し、それによって、複数の当該データ構造を生じる。

【0037】次に図2および9を参照して、スレーブリミットタイムスケジュールプロセス116を説明する。スレーブプロセス116は、ある時間実行し、終了する。このプロセスは周期的に起動する。図2に示す本発明の実施形態において、各リモートコピーペア108が関係するシーケンス番号115を有することを想起されたい。実際、リモートコピーペア108において待機している書込みデータ113の各ブロックについて、関係するシーケンス番号115が存在する。従って、各リモートコピーペア108はシーケンス番号115のリストを持ち得る。さらに、リモートディスクシステム105が、それと関係する複数のリモートコピーペア108を有し得ることを想起されたい。従って、各リモートディスクシステム105は複数のシーケンス番号のリストを持ち得る。

【0038】各リモートコピーペア108について、スレーブリミットタイムスケジュールプロセス116は、そのリモートコピーペアのシーケンス番号のリストを検査する（ステップ901）。それは、連続している番号の最長の連続を見つけ、その連続から最大シーケンス番号を返す。例えば、リモートコピーペアが以下のシーケンス番号のリストを含むと仮定する。．．．，11，12，13，14，15，16，17，18，19，20，21，22，23，24，25，26，28，29

【0039】上記の例では、シーケンス番号27がまだ

受信されていないので、26以下であるシーケンス番号は連続している。従って、プロセスは、この連続の中で最高値のシーケンス番号である「26」を選択する。次に、プロセスは、その最高値の（最大の）シーケンス番号と関係するタイムスタンプ114を検索する。これが各リモートコピーペアについて繰り返され、タイムスタンプのリストを生じる。

【0040】このタイムスタンプのリストから、最早タイムスタンプが選択される（ステップ901）。各スレーブリミットタイムスケジュールプロセス116は、このようにして1個のそうした最早タイムスタンプを生成する。各スレーブプロセスからの最早タイムスタンプはその後、マスタリミットタイムスケジュールプロセス117に送られる（ステップ902、図1も参照）。各スレーブプロセスはその後、マスタリミットタイムスケジュールプロセス117からリミットタイム値118を受信するために待ち（ステップ903）、値をキャッシュ202に格納する（ステップ904、図1も参照）。

【0041】次に図2および10を参照して、マスタリミットタイムスケジュールプロセス117を説明する。このプロセスは、リモートディスクシステムがスレーブプロセス116のそれぞれからタイムスタンプ114を受信した時に起動する。マスタプロセス117は、受信されたタイムスタンプから最早タイムスタンプを選択し（ステップ1000）、選択されたタイムスタンプをリミットタイム118として各スレーブプロセス116に送る（ステップ1001）。

【0042】次に図2および11を参照して、R書込みデータデステージプロセス215を説明する。このプロセスは、リモートディスクシステム105が、リモートコピーペア108に関係する書込みデータ113を、そのリモートコピーペア108と関係するリモートディスク装置107にデステージした時に実行される。リモートディスクシステム104は、関係するタイムスタンプ114が最も早い候補書込みデータ113を選択する（ステップ1100）。その後それは、選択されたタイムスタンプを、マスタリミットタイムスケジュールプロセス117により規定されたリミットタイム118と比較する（ステップ1101）。選択されたタイムスタンプ114がリミットタイム118より遅ければ、リモートディスク装置105は暫時待機し（ステップ1102）、その後、処理をステップ1100で続ける。選択されたタイムスタンプ114がリミットタイム118に等しいか、またはそれより早い場合、リモートディスクシステム105は、その関係するリモートディスクアドレス208および位置決め情報209に従って候補書込みデータ113をデステージする（すなわち、ディスクに書込む）（ステップ1103）。書込みデータならびにその関係するタイムスタンプ、シーケンス番号、リモートディスクアドレス情報208および位置決め情報2

09は、キャッシュメモリ203から除去される（ステップ1104）。

【0043】以下の例が役立つであろう。以下を有すると仮定する。

書込みデータA、タイムスタンプ10:00

書込みデータB、タイムスタンプ10:02

書込みデータC、タイムスタンプ10:04

リミットタイム:10:01

【0044】この例において、R書込みデータデステージプロセスは書込みデータAを選択し、それは最早タイムスタンプ（10:00）を有する。次に、R書込みデータデステージプロセスは、そのタイムスタンプ（10:00）がリミットタイム（10:01）より早いので、書込みデータAをリモートディスク装置107へデステージする。書込みデータAのデステージングの後、書込みデータBが最早タイムスタンプを有する。しかし、書込みデータBのタイムスタンプ（10:02）よりリミットタイム（10:01）が早いので、書込みデータBはデステージされることができない。書込みデータBのデステージングは、リミットタイム118がスレーブリミットタイムスケジュールプロセス116およびマスタリミットタイムスケジュールプロセス117によって10:02より遅い時刻に更新された後に可能になる（図9および図10に示す）。

【0045】ここで図3を参照して本発明の別の実施形態を説明する。図3のシステム図は、図2に示したものと本質的に同じであるが、以下の相違がある。図3に示す本発明の実施形態では、ローカルディスクシステム104と関係づけられている各リモートディスクシステム105についてシーケンス番号115が存在する。図2では、各リモートコピーペア108についてシーケンス番号が存在することを想起されたい。しかし、図3に示す実施形態の場合、ローカルディスクシステム104は、少なくとも1個のリモートコピーペア108をそのローカルディスクシステムと共有する各リモートディスク104システムについて1個のシーケンス番号115を有する。従って、図3においてわかる通り、各リモートディスクシステムについて、シーケンスカウンタ300およびリモートディスクシステムアドレス301よりなるデータペアが存在する。従って、ローカルディスクシステムが2個のリモートディスクシステムと関係づけられている場合、ローカルディスクシステムに含まれる2個の当該データペアが存在するはずである。

【0046】図3に示す実施形態において、L書込みデータ受信プロセス210は、図2に示す実施形態に関連して説明したものと同一である。

【0047】ここで図2および12を参照して、図3に示す本発明の実施形態についてL書込みデータ送信プロセス211を説明する。少なくとも1個のリモートコピーペア108をこのローカルディスクシステム104と

共有する各リモートディスク装置について1個のL書込みデータ送信プロセス211が存在する。ローカルディスクシステム104は、対応するリモートディスクシステム105に属する書込みデータ113の全部から、タイムスタンプ114が時間的に最も早く、リモートディスクシステム105にまだ送信されていない書込みデータ113を選択する（ステップ1200）。ローカルディスクシステム104はその後、そのリモートディスクシステムに対応するシーケンスカウンタ300の現在値を、選択された書込みデータ133（すなわち、関係するタイムスタンプが時間的に最も早い書込みデータ）と関係するシーケンス番号115にコピーする。シーケンスカウンタ300が増分される（ステップ601）。次に、選択された書込みデータ113ならびにその関係するタイムスタンプ114、シーケンス番号115、リモートディスクアドレス情報208および位置決め情報209が、リモートディスクシステム105に送信される（ステップ602）。プロセスはその後、図6の説明に従って継続する。

【0048】図3に示す本発明の実施形態において、L書込み送信完了プロセス212は、図2に示す本発明の実施形態に関連して説明したものと同一である。

【0049】図3に示す実施形態において、リモートディスクシステム105は、リモートコピーペア108をリモートディスクシステム105と共有するすべてのローカルディスクシステム104について、シーケンスカウンタ300およびローカルディスクシステムアドレス302よりなるデータペアを含む。リモートディスクシステムと関係する各ローカルディスクシステムについて1個の当該データペアが存在する。

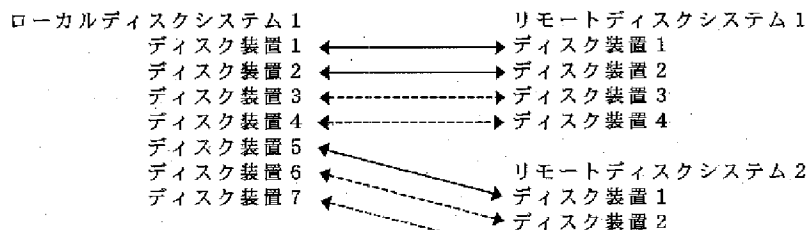
【0050】リモートコピーペア108をローカルディスクシステム105と共有するすべてのローカルディスクシステムについてR書込みデータ受信プロセス214が存在する。図3の実施形態において、R書込みデータ受信プロセス214は、図2に示す実施形態に関連して説明したものと同一である。同様に、マスタリミットタイムスケジュールプロセス117およびR書込みデータデステージプロセス215も図2に示す本発明の実施形態の説明と同じである。

【0051】ここで図13を参照して、図3に示す本発明の実施形態に従ったスレーブリミットタイムスケジュールプロセス116を検討する。各リモートディスクシステム105について1個のスレーブプロセスが存在する。このプロセスは、以下の相違はあるが、図2の実施形態に関して本質的に同一である。

【0052】少なくとも1個のリモートコピーペア108を共有する各ローカルディスクシステム104について、スレーブリミットタイムスケジュールプロセス116は、そのローカルディスクシステム104から受信されたシーケンス番号のリストを検査する（ステップ13

00)。プロセスは、連続であるシーケンス番号114の最大シーケンス番号を見つけ、その最大シーケンス番号に対応するタイムスタンプ115を返す。これは、タイムスタンプのリストを生成するために、少なくとも1個のリモートコピーペア108をリモートディスクシステムと共有する各ローカルディスクシステムについて繰り返される。このタイムスタンプのリスト11から、最早タイムスタンプが選択され（ステップ901）、マスタタイムリミットスケジュールプロセス117に送達される（ステップ902）。このようにして、各スレーブリミットタイムスケジュールプロセス116は、各リモートディスクシステム105について1個のそうした最早タイムスタンプを探索する。プロセスはその後、マスタリミットタイムスケジュールプロセス117からリミットタイム値118を受信するために待機し（ステップ903）、値をキャッシュ202に格納する（ステップ904）。

【0053】次に図4を参照して、本発明の別の実施形態を説明する。図4のシステム図は、以下の相違はある



【0055】図2～4に示す実施形態の間の相違は、ローカルディスクシステム104が7個のローカルディスク装置および2個のリモートディスクシステム105を有する、上記の例示的な機器構成に示されている。両矢印はリモートコピーペアを示す。例えば、ローカルディスク装置5およびリモートディスク装置1は、リモートコピーペア108を規定する。さらに、実線で描かれたリモートコピーペアは一体で1個のデータの完全性グループを規定するのに対し、破線で描かれたリモートコピーペアは一体で第2のデータの完全性グループを規定する。

【0056】従って、図2に示された本発明の実施形態によれば、ローカルディスクシステム104はそのディスク装置（合計7個のシーケンス番号）のそれぞれについて関係するシーケンス番号114を有するであろう。図4に示す実施形態によれば、ローカルディスクシステム104は、ローカルディスクシステム105と共通してリモートコピーペアを有する各リモートディスクシステム105について1個ずつ、2個の関係するシーケンス番号を有する。図4に示す実施形態によれば、ローカルディスクシステム104は、データの完全性ペアグループ109をローカルディスクシステム104と共有する各リモートディスクシステム105について1個ずつ、4個の関係するシーケンス番号114を有する。実

が、図2に関して示したものと本質的に同じである。図4に示す本発明の実施形態において、ローカルディスクシステムは、そのローカルディスクシステム104と共通してデータの完全性ペアグループ109を有する各リモートディスクシステム105について1個のシーケンス番号115を有する。例えば、ローカルディスクシステムXが、関係するデータの完全性ペアグループZを有するとする。データの完全性ペアグループはさらに、5個のリモートコピーペアグループRCP1～RCP5よりなる。さらに、リモートコピーペアグループRCP1およびRCP2はリモートディスクシステムAと関係し、リモートコピーペアRCP3はリモートディスクシステムBと関係し、RCP4およびRCP5はリモートディスクシステムCと関係するとみなす。この特定の例において、ローカルディスクシステムXは、データの完全性ペアグループZと関係する3個のシーケンス番号を有する。

【0054】

【表1】

線で描かれたデータの完全性ペアグループは2個のリモートディスクシステム104間で共有され、同様に、破線で描かれたデータの完全性ペアグループは2個のリモートディスクシステム104間で共有される。

【0057】さらに図4について続けられ、図示の通り、データの完全性ペアグループ109によってローカルディスクシステム104と関係づけられている各リモートディスクシステム105について、データの完全性ペアグループ情報データ構造400がローカルディスクシステム104に存在する。データ構造400は、データの完全性グループID401および、リモートディスクシステムアドレス403およびシーケンスカウンタ402よりなる1個以上のデータペアを含む。データの完全性ペアグループ109の各リモートディスクシステム105について1個の当該データペアが存在する。リモートディスクシステム105は、データの完全性ペアグループ108によってリモートディスクシステム105と関係づけられている各ローカルディスクシステム104について類似のデータ構造を備える。

【0058】図4に示す本発明の実施形態において、L書込み送信完了プロセス210およびL書込み送信完了プロセス212は、図2に示す本発明の実施形態に従って動作する。

【0059】次に図14を参照して、図4に示す本発明

の実施形態のL書き込みデータ送信プロセス211を説明する。少なくとも1個のデータの完全性ペアグループ109をそのローカルディスクシステム104と共有する各リモートディスク装置についてL書き込みデータ送信プロセス211が存在する。ローカルディスクシステム104は、対応するデータの完全性ペアグループ109に属する書き込みデータ113の全部から、タイムスタンプ114が時間的に最も早く、リモートディスクシステム105にまだ送信されていない書き込みデータ113を選択する(ステップ1400)。ローカルディスクシステムはその後、ターゲットリモートディスクシステムに対応するシーケンスカウンタ403の現在値を、選択された書き込みデータ113(すなわち、関係するタイムスタンプが時間的に最も早い書き込みデータ)と関係するシーケンス番号115にコピーする。シーケンスカウンタ403が増分される(ステップ601)。次に、選択された書き込みデータ113ならびにその関係するタイムスタンプ114、シーケンス番号115、リモートディスクアドレス情報208および位置決め情報209が、リモートディスクシステムに送られる(ステップ602)。プロセスはその後、図6に関する説明に従って継続する。

【0060】図4に示す本発明の実施形態において、R書き込みデータ受信プロセス214は、そのリモートディスクシステム105と関係するデータの完全性ペアグループ109を共有する各ローカルディスク装置104と関係づけられている。処理は図8に示すフローチャートに従って進行する。

【0061】マスタリミットタイムスケジュールプロセス117およびR書き込みデータデステージプロセス215は、図2に示す本発明の実施形態に関する説明と同様にして動作する。

【0062】次に図15を参照して、図4に示す本発明の実施形態に従ったスレーブリミットタイムスケジュールプロセス116を検討する。各リモートディスクシステム105について1個のスレーブプロセスが存在する。このプロセスは、以下の相違はあるが、図2の実施形態に関してと本質的に同じである。

【0063】データの完全性ペアグループによってリモートディスクシステムと関係づけられている各ローカルディスクシステム104について、スレーブリミットタイムスケジュールプロセス116は、そのローカルディスクシステムから受信されたシーケンス番号のリストを検査する(ステップ1500)。プロセスは、連続であるシーケンス番号114の最大シーケンス番号を見つけ、最大シーケンス番号に対応するタイムスタンプ115を返す。これは、タイムスタンプのリストを生成するために、データの完全性ペアグループ109によってリモートディスクシステムと関係する各ローカルディスクシステム104について繰り返される。タイムスタンプ

のこのリストから、最早タイムスタンプが選択され(ステップ901)、マスタタイムリミットスケジュールプロセス117に送達される(ステップ902)。このようにして、各スレーブリミットタイムスケジュールプロセス116は、各リモートディスクシステム105について1個のそうした最早タイムスタンプを探索する。プロセスはその後、マスタリミットタイムスケジュールプロセス117からリミットタイム値118を受信するために待機し(ステップ903)、値をキャッシュ202に格納する(ステップ904)。

【0064】ここで、リモートサイトに磁気テープ(MT)記憶装置システムが配備された状況における本発明の実施形態の検討のために図16に話を移す。ローカルコンピュータシステム1600は、少なくともローカルHOSTシステム1602および、ローカルディスクシステムといった少なくともローカル記憶装置システム1604を含む。ローカルコンピュータシステムはスイッチシステム1606を含み得る。少なくともリモートMTシステム1605を含むリモートコンピュータシステム1601が存在する。リモートHOSTシステム1603は、必ずしもリモートコンピュータシステム1601に存在するわけではない。

【0065】ローカルHOSTシステム1601、ローカルディスクシステム1604およびスイッチシステム1606は、通信路1613によってリモートディスクシステム1605と接続される。通信路1613は、その送信データパケットが必ずしもそれらが送られた順序で着信するわけではないことを特徴とするネットワークである。例えば、IPベースのネットワークはこの振舞いを呈する。本発明のこの実施形態において、読取り/書き込みデータの単位をブロックと称する。各ブロックは、自身を識別するために使用されるブロックIDを有する。

【0066】ローカル記憶装置システム1604は、少なくとも1個のローカル記憶装置1607および、キャッシュメモリ1611を備える記憶制御装置1609を有する。リモートMTシステム1605は、少なくとも1個のリモートMT装置1608および、キャッシュメモリ1611を備えるMT制御装置1610を含む。ローカルHOSTシステム1602はメインメモリ1612を有する。スイッチシステム1606はキャッシュメモリ1611も備える。

【0067】図16に示す本発明の実施形態によれば、リモートMT装置1608への読出し要求および書き込み要求はそれぞれ、読み出されるデータ1624または書込まれるデータ1625に伴うシーケンス番号1614を含む。

【0068】図16はまた、ローカルHOSTシステム1602、ローカルディスクシステム1604、リモートMTシステム1605およびスイッチシステム160

6のプロセスも示している。読出し要求発行プロセス1615および書込み要求発行プロセス1616は、ローカルHOSTシステム1602、ローカルディスクシステム1604およびスイッチシステム1606において提供される。リモートMTシステム1605は、読出し要求受信プロセス1617および書込み要求受信プロセス1618を含む。

【0069】読出し要求発行プロセス1615の処理は図17に概説されている。このフローは、ローカルHOSTシステム1602、ローカル記憶装置システム1604およびスイッチシステム1606の間で共通である。ローカル記憶制御装置1609のキャッシュメモリ1611に（またはメインメモリ1612に）コピーされるシーケンスカウンタ1623が、読取り／書込み要求が発行された時に、シーケンス番号1614の値を付与する（ステップ1700）。プロセスはその後、読出し要求をリモートMTシステム1605に発行する（ステップ1701）。プロセスはその後、シーケンスカウンタ1623を増分する（ステップ1702）。

【0070】次に（ステップ1703）、読出し要求発行プロセスは、発行された要求の数を確認する。その数が値m未満であれば、処理は、次の要求を発行するためにステップ1700に継続する。mの値は一般に>2であり、好ましくは経験的に決定される。その目的は、現在の要求の完了前に次の要求を送信することによってより良好な性能を得ることである。

【0071】数がm未満でなければ、プロセスは、リモートMTシステム1605からのデータ転送通知を待つ（ステップ1704）。読出し要求発行プロセス1615がリモートMTシステム1605からのデータ転送通知を受信すると、それは読出しデータ1624およびシーケンス番号1614を受信し、シーケンスカウンタ1623に従ってそれをキャッシュメモリ1611に（またはメインメモリ1612に）格納する（ステップ1705）。次に、読出し要求発行プロセス1615は、リモートMTシステム1605から全部のデータ転送通知を受信したかどうかを確認する（ステップ1706）。否定であれば、プロセスは、データ転送通知を待つためにステップ1703に継続する。

【0072】書込み要求発行プロセス1616の処理は、図18に示すフローチャートに概説されている。このフローは、ローカルHOSTシステム1602、ローカル記憶装置システム1604およびスイッチシステム1606間で共通である。書込み要求発行プロセスは、シーケンス番号1614に対するシーケンスカウンタ1623の内容をキャッシュメモリ1611に（またはメインメモリ1612に）コピーし（ステップ1800）、シーケンス番号1614および書込みデータ1625を伴う書込み要求をリモートMTシステム1605に発行する（ステップ1801）。その後、シーケンス

カウンタ1623を増分する（ステップ1802）。ステップ1803において、書込み要求発行プロセス1616は、発行された要求の数を確認する。その数が値m未満であれば、次の要求を発行するためにステップ1800にジャンプする。そうでなければ、プロセスはリモートMTシステムからのデータ転送通知を待つ（ステップ1804）。書込み要求発行プロセス1616がリモートMTシステム1605から通知を受信すると（ステップ1805）、プロセスはリモートMTシステム1605から全部のデータ転送通知を受信したかどうかを確認する（ステップ1806）。否定であれば、データ転送通知を待つためにステップ1803にジャンプする。

【0073】リモートMTシステム1605の読出し要求受信プロセス1617のフローは、図19に示す。リモートMTシステム1605は、MT制御装置1610のキャッシュメモリ1611に読出しデータ1624を有する。1群の読出しデータ1624が、1個のリモートMT装置1608から読み出され、MT待ち行列1626に格納される。各待ち行列エントリは、読出しデータ1624およびその対応するブロックID情報1627よりなるデータペアである。読出しデータ1624およびそのブロックID情報1627は、ブロックID情報1627に従って待ち行列に挿入される。従って、MT待ち行列1626はブロックID情報1627によって順序づけられる。この順序づけは、データブロックが要求側ローカルHOST1601に返送されるべき順序を示している。従って、一連の読出し要求は、適切な順番で受信されていれば、単に各データブロックを、それらがMT待ち行列1626に存在する順序で送ることによって満たされるであろう。

【0074】待ち行列1626は、二重連係データ構造である。それゆえ、各待ち行列エントリはさらに、次のブロックID情報1627を含む待ち行列エントリをポイントする正方向ポインタ1630および、以前のブロックID情報1627を含む待ち行列エントリをポイントする逆方向ポインタ1631を含む。ヘッドポインタ1628がMT待ち行列1626の先頭をポイントし、テールポインタ1629がMT待ち行列1626の末尾をポイントする。次のシーケンス番号カウンタ1632は、次の読出し要求によりアクセスされるべき読出しデータ1624に対応するシーケンス番号1614を含む。それゆえ、カウンタ1632の現在値は、MT待ち行列の先頭のデータブロックに対応する。

【0075】動作時、読出し要求受信プロセス1617は、ローカルHOST1601からシーケンス番号1614とともに読出し要求を受信する（ステップ1900）。読出し要求受信プロセスは、受信されたシーケンス番号1614を次のシーケンス番号カウンタ1632と比較する（ステップ1901）。それが等しい場合、プロセスは、受信されたシーケンス番号1614およ

び、（ヘッドポインタ1628によってポイントされる）MT待ち行列1626の先頭の読出しデータ1624を、受信されたシーケンス番号1614とともに、読出し要求発行プロセス1615へ送る（ステップ1902）。シーケンス番号カウンタ1632はその後更新され、MT待ち行列の次の読出しデータ1624のブロックID情報1627を参照する（ステップ1903）。プロセスはその後ステップ1907に継続する。

【0076】受信されたシーケンス番号が次のシーケンス番号カウンタ1632と等しくない場合、それは、乱順の読出し要求が受信されたことを意味する。ステップ1904において、読出し要求受信プロセス1617は、要求されている読出しデータのブロックID情報1627を計算する。詳細には、読出し要求受信プロセスは、次のシーケンス番号カウンタ1632と受信されたシーケンス番号1614との間の差Dを取得し、その差DをMT待ち行列1626の先頭の読出しデータ1624のブロックID情報Bに加算する。すなわち、 $B+D$ は見つけるべきブロックID情報1627である。ステップ1905において、読出し要求受信プロセスは、ブロックID情報1627が $B+D$ である待ち行列エントリを探索する。待ち行列1626のそのエントリにある読出しデータ1624は、受信されたシーケンス番号1614とともに、読出し要求発行プロセス1615に送られる（ステップ1906）。処理はステップ1907に続く。

【0077】ステップ1907において、読出し要求受信プロセス1617は、送信された読出しデータ1624に対応する待ち行列エントリを削除する。このように、シーケンス番号1614を使用することによって、リモートMTシステム1605は、たとえ読出し要求が通信路1613の性質のため乱順になった場合でも間違いなくアクセスされた読出しデータ1624を認識することができる。図17に戻って言えば、受信データは、受信されたシーケンス番号1614に従って組み立てられる。従って、リモートシステムにおける読出し要求が乱順になった場合、シーケンス番号は、それらが正しい順序で満たされるように保証する。同様に、ローカルシステムにおいて受信データが乱順になった場合も、それはシーケンス番号1614により正しく組み立てられる。

【0078】リモートMTシステム1605における書き込み要求受信プロセス1618のフローは、図20に示す。書込まれるデータのMT待ち行列1626の構造は、読出しデータ1624のそれと同じである。書き込みデータ1625の場合、MT装置1605の次のシーケンス番号カウンタ1632は、テールポインタ1629によってポイントされる書き込みデータのシーケンス番号1614よりも1だけ大きい。

【0079】リモートコンピュータにおける書き込み要求

受信プロセス1618は、シーケンス番号1614および書き込みデータ1625を伴う書き込み要求を受信し、書き込みデータをキャッシュメモリ1611に格納する（ステップ2000）。次に、書き込み要求受信プロセス1618は、シーケンス番号1614が次のシーケンス番号カウンタ1632と等しいかまたはそれより大きいかどうかを確認する（ステップ2001）。等しければ、それは対応する書き込みMT待ち行列1626の末尾の受信された書き込みデータ1625を挿入し、従ってMT待ち行列の情報を更新する（ステップ2002）。プロセスはその後、受信された書き込みデータ1625のブロックIDを計算し、それをブロックID情報1627に格納する（ステップ2003）。その後、 $D2$ （シーケンス番号1614と次のシーケンス番号カウンタ1632との差）+1を次のシーケンス番号カウンタ1632に加算し（ステップ2004）、ステップ2009にジャンプする。

【0080】受信されたシーケンス番号が次のシーケンス番号カウンタ1632より小さい場合、プロセスは、次のシーケンス番号カウンタ1632と受信されたシーケンス番号1614との差を計算する（ステップ2005）。プロセスは、書き込みデータ1624を挿入するためのMT待ち行列1626の位置を見つけ、従ってMT待ち行列の情報を更新する（ステップ2007）。その後プロセスは、受信された書き込みデータ1625のブロックIDを計算し、それをブロックID情報1627に格納する（ステップ2008）。次に、書き込み要求受信プロセスは書き込み要求の完了を通知し、受信されたシーケンス番号1614を書き込み要求発行プロセス1616に返送する（ステップ2009）。このようにして、シーケンス番号1614を使用することによって、リモートMTシステム1605はやはり、たとえ書き込み要求が通信路1613のために乱順になった場合でも間違いなく書き込みデータ1625のブロックIDを認識することができる。

【0081】図21は、MT記憶装置システムに関する本発明の別の実施形態を示す。本発明のこの実施形態は、図16に示す実施形態と本質的に同じである。相違は、乱順の発生を検出し、当該発生を訂正するために、図16において使用されたシーケンス番号1614の代わりに次のブロックID情報1700、1701を使用することである。

【0082】

【発明の効果】本発明によれば、信頼できるIPベースのデータ回復システムを提供することができる。

【図面の簡単な説明】

【図1】本発明の一般化したシステム図である。

【図2】シーケンス番号がリモートコピーペアごとに割り当てられる本発明の1実施形態の一般化したブロック図である。

【図３】シーケンス番号がリモートディスクシステムごとに維持される本発明の別の実施形態の一般化したブロック図である。

【図４】シーケンス番号がデータの完全性ペアグループごとに管理される本発明の別の実施形態の一般化したブロック図である。

【図５】本発明の１実施形態に従ったプロセスフローを概説する。

【図６】本発明の１実施形態に従ったプロセスフローを概説する。

【図７】本発明の１実施形態に従ったプロセスフローを概説する。

【図８】本発明の１実施形態に従ったプロセスフローを概説する。

【図９】本発明の１実施形態に従ったプロセスフローを概説する。

【図１０】本発明の１実施形態に従ったプロセスフローを概説する。

【図１１】本発明の１実施形態に従ったプロセスフローを概説する。

【図１２】図６に示すプロセスフローの代替実施形態を示す。

【図１３】図９に示すプロセスフローの代替実施形態を示す。

【図１４】図６に示すプロセスフローの別の代替実施形態を示す。

【図１５】図９に示すプロセスフローの別の代替実施形態を示す。

【図１６】本発明の磁気テープ記憶装置システムの実施形態を示す。

【図１７】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。

【図１８】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。

【図１９】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。

【図２０】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。

【図２１】本発明の別の磁気テープ実施形態である。

【符号の説明】

１００…ローカルコンピュータシステム、１０１…リモートコンピュータシステム、１０２…ローカルHOSTシステム、１０３…リモートHOSTシステム、１０４…ローカルディスクシステム、１０５…リモートディスクシステム、１０６…ローカルディスク装置、１０７…リモートディスク装置、１０８…リモートコピーペア、１０９…データの完全性ペアグループ、２００…ローカルディスク制御装置、２０１…リモートディスク制御装置、２０２…キャッシュメモリ、４０２…シーケンスカウンタ、１６０５…リモートMTシステム、１６０６…スイッチシステム、１６０８…リモートMT装置、１６１０…MT制御装置、１６１２…メインメモリ。

Storage System Connected to a Data Network With Data Integrity

BACKGROUND OF THE INVENTION

The present invention relates generally to data storage systems and more particularly to maintaining data integrity of data storage systems in a data network environment.

Conventionally, data processing systems have access to their associated data storage systems over a high speed, high reliability data bus. However, opportunities become available as the widespread use of network communications continues to expand. IP (Internet Protocol) provides the basic packet delivery service on which TCP/IP (transport control protocol/IP) networks are built. IP is a well defined protocol and is therefore a natural candidate for providing the transport/networking layer for network-based data storage access, where server systems exchange data with storage systems and storage systems exchange data with other storage systems using IP.

The nature of IP, however, presents some unique problems in the area of data storage access systems. First, IP is a connectionless protocol. This means that IP does not exchange control information to establish an end-to-end connection prior to transmitting data. IP contains no error detection and recovery mechanism. Thus, while IP can be relied on to deliver data to a connected network, there is no mechanism to ensure the data was correctly received or that the data is received in the order that it was sent. IP relies on higher layer protocols to establish the connection if connection-oriented service is desired.

In a data storage system where dual remote copy capability is needed, IP-based transmission presents a problem. A remote copy function provides a real time copy of a primary data store at a remote site with the goal of realizing disaster recovery in the primary data store. It is important to guarantee data integrity in order that this function serves its purpose. There are two types of remote copy: synchronous and asynchronous.

In a synchronous type remote copy, a write request by a local HOST to its associated local disk system does not complete until after the written data is transferred from the local disk system to a remote disk system. Thus, in the case of synchronous type copy, it is easy to ensure data integrity between the local and the remote disk system.

In an asynchronous type remote copy, a write request by the local HOST completes before the local disk completes its transfer to the remote disk. As the name

implies, control is returned to the local HOST irrespective of whether the transfer operation from the local disk to the remote disk completes. Data integrity during an asynchronous type copy operation, therefore, relies on a correct arrival order of data at the remote disk system so that data on the remote disk is written in the same order as on the
5 local disk.

To achieve this, the local disk system includes a time stamp with the data that is sent to the remote disk. Data at the remote is written according to the order of the time stamp. Thus, for example, when the remote disk receives data with a time stamp 7:00, it has already received all data whose time stamps precede 7:00.

10 However, in an IP-based network, when packets can arrive out of sequence, a data packet having a time stamp of 7:00 may or may not be preceded by data packets having an earlier time stamp. Consequently, it is difficult to ensure data integrity at a remote disk system when the transmission protocol is based on connectionless transport model such as the IP.

15 Another problem arises when IP is used with magnetic tape systems. Read and write operations to magnetic tape is sequential and so the addressing is fixed. Thus, in the case where a data packet arrives at the remote site out of sequence, the data will be written to tape in incorrect order. A subsequent recovery operation from tape to restore a crashed storage system would result in corrupted data.

20 There is a need to provide a reliable IP-based data recovery system.

SUMMARY OF THE INVENTION

A data storage system in accordance with the invention comprises a local storage component and a remote storage component. Data to be written to the local
25 storage component is sent in a data packet to the remote storage component over a data network.

The data packet includes a copy of the data to be written at the local storage component, a time stamp, and a sequence number. Plural such data packets are received at the remote storage component. The data packets are selected for writing at the
30 remote storage component based on the sequence numbers and the time stamps associated with each data packet.

In one embodiment of the invention, the local and remote storage components are configured as plural local and remote disk units, respectively. Each local

disk unit is associated with a remote disk unit. Each such pairing is called a remote copy pair. In this embodiment, each remote copy unit has an associated sequence number.

In another embodiment of the invention, the local disk units are grouped into local disk systems. Similarly, each remote disk unit is grouped into remote disk systems. In this embodiment of the invention, there is a sequence number for each pair of local and remote disk systems which have at least one common remote copy pair.

In another embodiment of the invention, the remote copy pairs are grouped into data integrity pair groups. In this embodiment, there is a sequence number for each pair of local and remote disk systems which have in common at least one data integrity pair group.

BRIEF DESCRIPTION OF THE DRAWINGS

The teachings of the present invention can be readily understood by considering the accompanying detailed description in conjunction with the following drawings:

Fig. 1 is a generalized system diagram of the present invention;

Fig. 2 is a generalized block diagram of one embodiment of the invention in which sequence numbers are allocated on a per remote copy pair basis;

Fig. 3 is a generalized block diagram of another embodiment of the invention in which sequence numbers are maintained on a per remote disk system basis;

Fig. 4 is a generalized block diagram of another embodiment of the invention in which sequence numbers are managed on a per data integrity pair group basis;

Figs. 5 – 11 outline process flows in accordance with one embodiment of the invention;

Fig. 12 shows an alternate embodiment of the process flow shown in Fig. 6;

Fig. 13 shows an alternate embodiment of the process flow shown in Fig. 9;

Fig. 14 shows another alternate embodiment of the process flow shown in Fig. 6;

Fig. 15 shows another alternate embodiment of the process flow shown in Fig. 9;

Fig. 16 shows a magnetic tape storage system embodiment of the present invention;

Figs. 17 – 20 show process flows in an embodiment of the invention in a magnetic tape storage system; and

5 Fig. 21 is another magnetic tape embodiment of the invention.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

Referring to Fig. 1, a computer system in accordance with one embodiment of the invention is shown. A local computer system 100 includes at least a
10 local HOST system 102 and at least a local disk system 104. The remote computer system 101 includes at least a remote disk system 105. A remote HOST system 103 is not always necessary for the remote computer system 101. Each local disk system 104 is connected to a remote disk system 105 through a communication path 112. The
15 communication path 112 is a network which is characterized in that transmitted data packets do not necessarily arrive in the order in which they were sent. An IP-based network exhibits this behavior. In general, a connectionless network exhibits this behavior. As an example, a wide area network (WAN) can be based on IP. The invention, however, is not limited to WAN's.

The local disk system 104 includes at least a local disk unit 106 which has
20 an associated real time copy maintained in the remote disk system 105. The remote disk system 105 includes at least a remote disk unit 107 which contains a real time copy of a local disk unit 106. The pair comprising a local disk unit 106 and a remote disk unit 107 is called a remote copy pair 108. A group of remote copy pairs 108, among which data integrity must be guaranteed, is called a data integrity pair group 109. A group of local
25 disk units 106 which belong to a data integrity pair group 109 is called a data integrity local disk group 110. Similarly at the remote system, a group of remote disk units 107 which belong to one data integrity pair group 109 is called a data integrity remote disk group 111.

The data integrity local disk group 110 may comprise local disk units 106
30 from a single local disk system 104, or from two or more local disk systems 104. Consequently, the constituent local disk units 106 of a local disk system 104 may found in one or more data integrity pair groups 109. The data integrity remote disk unit group 111 may comprise remote disk units 107 which belong to one or more remote disk

systems 105. Consequently, a remote disk system 105 may have remote disk units 107 which belong to different data integrity pair groups 109.

During the course of operation of the local HOST system 102, data will need to be written to the local disk system 104. The local HOST will transfer “write data” 113 to be stored on the local disk system 104. The local disk system 104 also sends the write data 113 to a remote disk system 105 for data recovery. In accordance with the invention, when the local disk system 104 sends the write data 113 to be written to the remote disk system 105, it also sends time stamp 114 and a sequence number 115. The time stamp 114 shows the time when the local disk system received the request from the local HOST system 102. The sequence number 115 is the sequence number of the write data 113. The sequence number 115 is generated when a local disk unit 104 sends write data 113 to a remote disk system 105. The local disk system 104 selects the sequence number 114 and ensures that they are sequentially generated.

To realize data integrity, the order of data written on the remote disk units 107 in a data integrity pair group 109 needs to be same as the order of the same data written on the corresponding local disk units 106 of that data integrity pair group. To guarantee the data integrity in the data integrity pair group 109, it is necessary to compare the time stamps 114 from among the remote disk systems 105 because it is possible that one of the remote disk systems 105 will have received write data 113 with a time stamp 114 of 7:00, while another of the remote disk systems 105 has yet to receive write data 113 with an earlier time stamp.

Thus, the remote system 101 includes plural slave limit time schedule processes 116. Each remote disk system 105 has an associated slave limit time schedule process 116. The remote system further includes a single master limit time schedule process 117. The slave limit time schedule processes 116 each send information for time stamp 114 to the master limit time schedule process 117. The master limit time schedule process 117 then decides the earliest time to permit de-staging to a remote disk unit 107. This earliest time is sent as a time limit 118 to each slave time limit scheduling process 116.

In accordance with the invention, the local disk system 104 sends a sequence number 115 with each packet of write data 113 to ensure data integrity in the remote copy system, even in the event that the data packets arriving at the remote system arrive out of order.

Referring to Fig. 2, in an embodiment of the invention, the figure shows the processes in the local disk system and the remote disk system where every remote copy pair 108 has one sequence number. The local disk system 104 includes a local disk control unit 200 having a cache memory 202. The remote disk system 105 includes a
5 remote disk control unit 201 with a cache memory 203.

The processes include an L-write data receive process 210, an L-write data send process 211, an L-write data send completion process 212, and an L-write data destage process on the local computer system 100. The remote computer system 101 includes an R-write data receive process 214, an R-write data destage process 215, and
10 the master and slave limit time scheduling processes, 117, 116 respectively. Each process is activated in its respective local disk control unit 200 or remote disk control unit 201.

For each remote copy pair 108, there is a remote copy pair data store 203 which includes a local disk address 204, a remote disk address 205, a data integrity group id 206, and a sequence counter 207. The local disk address 204 and the remote disk
15 address 205 are the destination addresses respectively of the local disk unit and remote disk unit which comprise the remote copy pair 108, for a write operation. The data integrity pair group id 206 identifies to which data integrity pair group 109 the remote copy pair belongs. As shown in Fig. 2, each remote copy pair 108 has an associated sequence number 115. The sequence counter 207 provides the sequence number 115 that
20 is associated with the write data 113 that is sent to the remote disk.

The cache memory 202 also contains, for each remote copy pair 108, a data structure comprising write data 113, a remote disk address 208, positioning information 209, a sequence number 115, and a time stamp 114. Thus, there is one such data structure for each block of write data that is sent to the local disk system from the
25 local HOST 102, thereby resulting in plural such data structures for a typical write operation by local HOST 102. The positioning information is, for example, the disk block address on disk drives commonly found in personal computers. In conventional mainframe systems, the positioning information is typically the cylinder number, head number, and record number.

30 Referring now to Figs. 2 and 5, the processing of the L-write data receive process 210 will be described. This process is executed in the local disk system when it receives a write request from the local HOST system 102. First, the local disk system 104 receives a write request from the local HOST system 102 (step 500). A write request specifies the address information of the local disk unit 106 and the position on the local

disk unit 106 to which the write data 113 is written. Next, the local disk system 104 receives the actual write data 113 to be written and caches it in cache memory 202 (step 501). The local disk system 104 obtains the remote disk address 205 of the corresponding remote disk unit 107 in the remote copy pair 108 from the remote copy pair data store 203 and stores it in the cache memory as remote disk address 208. The position specified in the write request is stored in the cache memory as position information 209 (step 502). The local disk system then receives a time stamp from the local HOST system and stores it into the cache memory 202 as time stamp 114 (step 503). The time stamp 114 can be generated other than by the local HOST system 104, so long as the time stamp 114 that is produced is common to all of the local disk systems 104. In an embodiment where there are more than one local HOST systems 102, a shared clock is assumed to exist which provides a common time stamp value amongst the HOST systems. Finally, the local disk system 104 indicates to the local HOST system 102 the completion of the write request (step 504).

Referring to Figs. 2 and 6, the processing of the L-write data send process 211 will be described. This process is executed when the local disk system is ready to send write data 113 to the remote disk system 105. In accordance with the embodiment of the invention shown in Fig. 2, there is an L-write data send process 211 for each remote copy pair 108. This process is executed asynchronously relative to the L-write data receive process 210.

The local disk system 104 selects the write data 113 whose associated time stamp 114 is the earliest in time from among all of the write data 113 that are waiting in the remote copy pair 108 to be sent to the remote disk system (step 600). The local disk system then takes the current value of the sequence counter 207 as the sequence number 115 that will be associated with the selected write data 113 (i.e., the write data whose associated time stamp is the earliest in time). The sequence counter 207 is incremented (step 601). Next, the selected write data 113 and its associated time stamp 114, sequence number 115, remote disk address information 208, and position information 209 are sent to the remote disk system 105 (step 602, see also Fig. 1).

In accordance with the invention, the L-write data send process 211, then proceeds to process the next write data 113 without waiting for any indication from the remote disk system 105 as to completion. In this manner, high data transfer rates are realized. However, there is the possibility of packets arriving out of sequence. Hence, the local disk system 104 checks whether there are any write data 113 that belong to the

remote disk system 105 in the remote copy pair 108 which have not yet been sent (step 603). If so, processing continues at step 600. If not, then the process waits for a while (step 604) and then continues at step 603.

5 The waiting in step 604 is to accommodate the situation where there is no write data to be sent. If there is no write data to send in step 603, the L-write data send process 211 has no processing to do. Consequently, step 604 is used to pause the process for a while before checking again to see if there is write data to send.

Referring now to Figs. 2 and 7, the processing in the L-write data send completion process 212 will be described. This process is executed when the local disk system 104 receives a notification of the completion of the transfer of the write data 113 to the remote disk system 105. The local disk system receives the notification of the transfer of the write data 113 from the remote disk system 105 (step 700). The local disk system 104 then makes the value of the corresponding time stamp 114 and the sequence number 115 NULL. Since the write data 113 has already been sent to the remote disk system 105, these fields are no longer used by the local disk system 104.

After the transfer of write data 113 to the remote disk system 104 is completed, the write data 113 is then written to the local disk unit 104 by the L-write data destage process 213. This process results in the actual writing of the write data 113 to the local disk unit 106 and is performed according to known conventional methods.

20 Referring now to Figs. 2 and 8, the R-write data receive process 214 will be discussed. This process is executed in the remote disk system 105 when it receives write data 113 from the local disk system 104. In accordance with the embodiment of the invention shown in Fig. 2, there is an R-write data receive process 214 for each remote copy pair 108. The remote disk system 105 stores the received write data 113 and its associated time stamp 114, sequence number 115, remote disk address information 208, and position information 209 into cache memory 203 (step 800). The remote disk system 105 sends a completion indication of the transfer of the write data 113 along with its sequence number 115 back to the local disk system (step 801).

30 The cache memory 203 contains, for each remote copy pair 108, a data structure comprising a received write data 113, a remote disk address 208, positioning information 209, a sequence number 115, and a time stamp 114. Thus, there is one such data structure for each block of write data 113 that has been received from the local HOST 102, thereby resulting in plural such data structures.

Referring now to Figs. 2 and 9, the slave limit time schedule process 116 will be described. The slave process 116 executes for a period of time and terminates. This process is activated periodically. Recall that in the embodiment of the invention shown in Fig. 2, each remote copy pair 108 has an associated sequence number 115. In fact, for each block of write data 113 that is waiting in a remote copy pair 108, there is an associated sequence number 115. Thus, each remote copy pair 108 may have a list of sequence numbers 115. Recall further that a remote disk system 105 may have more than one remote copy pair 108 associated with it. Consequently, each remote disk system 105 may have more than one list of sequence numbers.

For each remote copy pair 108, the slave limit time schedule process 116 inspects the list of sequence numbers in that remote copy pair (step 901). It finds the longest run of numbers that are sequential and returns the maximum sequence number from that run. For example, suppose a remote copy pair contained the following list of sequence numbers:

..., 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29

In the above example, the sequence numbers which are 26 or less than 26 are sequential because the sequence number 27 has not been yet received. And so the process will select '26' which is the highest-valued sequence number in the run. Next, the process searches the time stamp 114 associated with the highest-valued (maximum) sequence number. This is repeated for each remote copy pair, resulting in a list of time stamps.

From this list of time stamps, the earliest time stamp is selected (step 901). Each slave limit time schedule process 116 produces one such earliest time stamp in this manner. The earliest time stamp from each slave process is then delivered to the master time limit schedule process 117 (step 902, see also Fig. 1). Each slave process then waits (step 903) to receive a limit time value 118 from the master limit time schedule process 117 and stores the value in cache 202 (step 904, see also Fig. 1).

Referring now to Figs. 2 and 10, the master limit time schedule process 117 will be described. This process is activated when the remote disk system receives time stamps 114 from each of the slave processes 116. The master process 117 selects the earliest time stamp from among the received time stamps (step 1000) and sends the selected time stamp as a limit time 118 to each slave process 116 (step 1001).

Referring now to Figs. 2 and 11, the R-write data destage process 215 will be described. This process is executed when the remote disk system 105 destages write data 113 associated with a remote copy pair 108 onto the remote disk unit 107 associated with that remote copy pair 108. The remote disk system 104 selects a candidate write data 113 whose associated time stamp 114 is the earliest (step 1100). Then it compares the selected time stamp with the limit time 118 (step 1101) defined by the master limit time schedule process 117. If the selected time stamp 114 is later than the limit time 118, then the remote disk unit 105 waits for a while (step 1102) and then continues processing at step 1100. If the selected time stamp 114 is equal to or earlier than the limit time 118, then the remote disk system 105 destages (i.e., writes out to disk) the candidate write data 113 according to its associated remote disk address 208 and position information 209 (step 1103). The write data and its associated time stamp, sequence number, remote disk address information 208, and position information 209 are removed from cache memory 203 (step 1104).

The following example will be helpful. Suppose we have the following:

```
write data A, time stamp 10:00
write data B, time stamp 10:02
write data C, time stamp 10:04
:
:
limit time: 10:01
```

In this example, the R-write data destage process selects write data A, which has the earliest time stamp (10:00). Next, the R-write data destage process destages write data A to a remote disk unit 107 because its time stamp (10:00) is earlier than the limit time (10:01). After the destaging of the write data A, write data B has the earliest time stamp. However, write data B cannot be destaged because the limit time (10:01) is earlier than the write data B's time stamp (10:02). The destaging of the write data B will become possible after the limit time 118 is updated to a time later than 10:02 by the slave limit time schedule process 116 and the master limit time schedule process 117 (shown in Fig. 9 and Fig. 10).

Referring now to Fig. 3, another embodiment of the invention will be described. The system diagram of Fig. 3 is essentially the same as that shown for Fig. 2, with the following differences. In the embodiment of the invention shown in Fig. 3, there is a sequence number 115 for each remote disk system 105 that is associated with the local disk system 104. Recall in Fig. 2, there is a sequence number for each remote copy

pair 108. However, for the embodiment shown in Fig. 3, the local disk system 104 has one sequence number 115 for each remote disk 104 system which shares at least one remote copy pair 108 with that local disk system. Thus, as can be seen in Fig. 3, there is a data pair comprising a sequence counter 300 and a remote disk system address 301, for each remote disk system. Thus, if a local disk system is associated with two remote disk systems, there will be two such data pairs contained in the local disk system.

In the embodiment shown in Fig. 3, the L-write data receive process 210 is the same as described in connection with the embodiment shown in Fig. 2.

Referring now to Figs. 2 and 12, the L-write data send process 211 for the embodiment of the invention shown in Fig. 3 will be described. There is an L-write data send process 211 for each remote disk unit which shares at least one remote copy pair 108 with this local disk system 104. The local disk system 104 selects the write data 113 whose time stamp 114 is the earliest in time from all of the write data 113 belonging to the corresponding remote disk system 105 and which have not yet been sent to the remote disk system 105 (step 1200). The local disk system 104 then copies the current value of the sequence counter 300 corresponding to that remote disk system into the sequence number 115 associated with the selected write data 113 (i.e., the write data whose associated time stamp is the earliest in time). The sequence counter 300 is incremented (step 601). Next, the selected write data 113 and its associated time stamp 114, sequence number 115, remote disk address information 208, and position information 209 are sent to the remote disk system 105 (step 602). The process then continues in accordance with the description for Fig. 6.

In the embodiment of the invention shown in Fig. 3, the L-write send completion process 212 is the same as described in connection with the embodiment of the invention shown in Fig. 2.

In the embodiment shown in Fig. 3, the remote disk system 105 includes a data pair comprising a sequence counter 300 and a local disk system address 302, for every local disk system 104 which shares a remote copy pair 108 with the remote disk system 105. There is one such data pair for each local disk system that is associated with the remote disk system.

There is an R-write data receive process 214 for every local disk system which shares a remote copy pair 108 with the local disk system 105. In the embodiment of Fig. 3, the R-write data receive process 214 is the same as described in connection with the embodiment shown in Fig. 2. Similarly, the master limit time schedule process 117

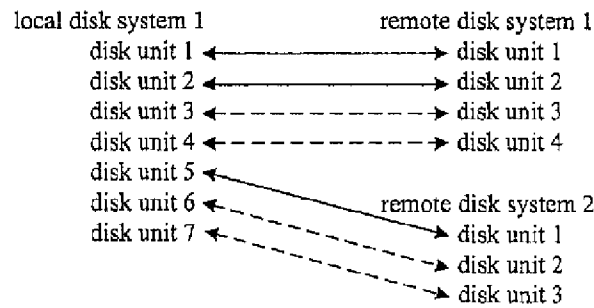
and the R-write data destage process 215 are the same as described for the embodiment of the invention shown in Fig. 2.

Referring now to Fig. 13, the slave limit time schedule process 116 in accordance with the embodiment of the invention shown in Fig. 3 will be discussed.

5 There is a slave process for each remote disk system 105. This process is essentially the same as for the embodiment of Fig. 2, with the following difference.

For each local disk system 104 that shares at least one remote copy pair 108, the slave limit time schedule process 116 inspects the list of sequence numbers received from that local disk system 104 (step 1300). It finds the maximum sequence
10 number of the sequence numbers 114 that are sequential and returns the time stamp 115 corresponding to the maximum sequence number. This is repeated for each local disk system that shares at least one remote copy pair 108 with the remote disk system to produce a list of time stamps. From this list of time stamps 11, the earliest time stamp is selected (step 901) and delivered to the master time limit schedule process 117 (step 902).
15 Thus, each slave limit time schedule process 116 searches one such earliest time stamp for each remote disk system 105. The process then waits (step 903) to receive a limit time value 118 from the master limit time schedule process 117 and stores the value in cache 202 (step 904).

Referring now to Fig. 4, another embodiment of the invention will be
20 discussed. The system diagram of Fig. 4 is essentially the same as that shown for Fig. 2, with the following differences. In the embodiment of the invention shown in Fig. 4, local disk system has a sequence number 115 for each remote disk system 105 that has a data integrity pair group 109 in common with that local disk system 104. Consider, for example, that a local disk system X has an associated data integrity pair group Z. The
25 data integrity pair group in turn comprises five remote copy pair groups, RCP1 – RCP5. Consider further that remote copy pair groups RCP1 and RCP2 are associated with remote disk system A, remote copy pair RCP3 is associated with remote disk system B, and RCP4 and RCP5 are associated with remote disk system C. In this particular example, the local disk system X will have three sequence numbers associated with data
30 integrity pair group Z.



The differences among the embodiments shown in Figs. 2 – 4 are illustrated in the exemplary configuration shown above, where a local disk system 104 has seven local disk units and two remote disk systems 105. The double arrow indicates a remote copy pair; e.g. local disk unit 5 and remote disk unit 1 define a remote copy pair 108. In addition, the solid-lined remote copy pairs together define one data integrity group, while the broken-lined remote copy pairs together define a second data integrity group.

Thus, in accordance with the embodiment of the invention shown in Fig. 2, the local disk system 104 would have an associated sequence number 114 for each of its disk units (a total of seven sequence numbers). In accordance with the embodiment shown in Fig. 4, the local disk system 104 has two associated sequence numbers, one for each remote disk system 105 which has a remote copy pair in common with the local disk system 105. In accordance with the embodiment shown in Fig. 4, the local disk system 104 has four associated sequence numbers 114, one for each remote disk system 105 that shares a data integrity pair group 109 with the local disk system 104. The solid-lined data integrity pair group is shared between the two remote disk systems 104; likewise, the broken-lined data integrity pair group is shared between the two remote disk systems 104.

Now continuing with Fig. 4, as can be seen, there is a data integrity pair group information data structure 400 in the local disk system 104 for each remote disk system 105 that is associated with the local disk system 104 via a data integrity pair group 109. The data structure 400 includes a data integrity group id 401 and one or more data pairs comprising a remote disk system address 403 and a sequence counter 402. There is one such data pair for each remote disk system 105 in the data integrity pair group 109. The remote disk system 105 includes a similar data structure, for each local disk system 104 that is associated with the remote disk system 105 via a data integrity pair group 108.

In the embodiment of the invention shown in Fig. 4, the L-write send completion process 210 and the L-write send completion process 212 operate in accordance with embodiment of the invention shown in Fig. 2.

Referring now to Fig. 14, the L-write data send process 211 for the
5 embodiment of the invention shown in Fig. 4 will be described. There is an L-write data send process 211 for each remote disk unit which shares at least one data integrity pair group 109 with this local disk system 104. The local disk system 104 selects the write data 113 whose time stamp 114 is the earliest in time from all of the write data 113 belonging to the corresponding data integrity pair group 109 and which have not yet been
10 sent to the remote disk system 105 (step 1400). The local disk system then copies the current value of the sequence counter 403 corresponding to the target remote disk system into the sequence number 115 associated with the selected write data 113 (i.e., the write data whose associated time stamp is the earliest in time). The sequence counter 403 is incremented (step 601). Next, the selected write data 113 and its associated time stamp
15 114, sequence number 115, remote disk address information 208, and position information 209 are sent to the remote disk system (step 602). The process then continues in accordance with the discussion for Fig. 6.

In the embodiment of the invention shown in Fig. 4, an R-write data receive process 214 is associated with each local disk unit 104 that shares a data integrity
20 pair group 109 associated with this remote disk system 105. Processing proceeds in accordance with the flow chart shown in Fig. 8.

The master limit time schedule process 117 and the R-write data destage process 215 operate in the same manner as described for the embodiment of the invention shown in Fig. 2.

Referring now to Fig. 15, the slave limit time schedule process 116 in
25 accordance with the embodiment of the invention shown in Fig. 4 will be discussed. There is a slave process for each remote disk system 105. This process is essentially the same as for the embodiment of Fig. 2, with the following difference.

For each local disk system 104 that is associated with the remote disk
30 system via a data integrity pair group, the slave limit time schedule process 116 inspects the list of sequence numbers received from that local disk system (step 1500). It finds the maximum sequence number of the sequence numbers 114 that are sequential and returns the time stamp 115 corresponding to the maximum sequence number. This is repeated for each local disk system 104 associated with the remote disk system via a data integrity

pair group 109 to produce a list of time stamps. From this list of time stamps, the earliest time stamp is selected (step 901) and delivered to the master time limit schedule process 117 (step 902). Thus, each slave limit time schedule process 116 searches one such earliest time stamp for each remote disk system 105. The process then waits (step 903) to
5 receive a limit time value 118 from the master limit time schedule process 117 and stores the value in cache 202 (step 904).

Turn now to Fig. 16 for a discussion of an embodiment of the invention in the context of magnetic tape (MT) storage systems deployed at the remote site. A local computer system 1600 includes at least a local HOST system 1602 and at least a local
10 storage system 1604 such as a local disk system. The local computer system may include a switch system 1606. There is a remote computer system 1601 that includes at least a remote MT system 1605. The remote HOST system 1603 is not always present in the remote computer system 1601.

The local HOST system 1601, the local disk system 1604, and the switch
15 system 1606 are connected to the remote disk system 1605 via a communication path 1613. The communication path 1613 is a network which is characterized in that transmitted data packets do not necessarily arrive in the order in which they were sent. For example, an IP-based network exhibits this behavior. In this embodiment of the invention, a unit of read/write data is called a block. Each block has a block id which is
20 used to identify itself.

A local storage system 1604 has at least one local storage unit 1607 and a storage control unit 1609 with a cache memory 1611. The remote MT system 1605 includes at least one remote MT unit 1608 and an MT control unit 1610 with a cache memory 1611. The local HOST system 1602 has a main memory 1612. The switch
25 system 1606 also includes a cache memory 1611.

In accordance with the embodiment of the invention shown in Fig. 16, a read request and a write request to a remote MT unit 1608 each includes a sequence number 1614 that accompanies the data to be read 1624 or written 1625.

Fig. 16 also shows the processes of the local HOST system 1602, the local
30 disk system 1604, the remote MT system 1605, and the switch system 1606. A read request issue process 1615 and a write request issue process 1616 are provided in the local HOST system 1602, the local disk system 1604, and the switch system 1606. The remote MT system 1605 includes a read request receive process 1617 and a write request receive process 1618.

Processing of the read request issue process 1615 is outlined in Fig. 17. This flow is common among the local HOST system 1602, the local storage system 1604 and the switch system 1606. A sequence counter 1623 that is copied to cache memory 1611 of the local storage control unit 1609 (or into main memory 1612) provides the value for sequence number 1614 when a read/write request is issued (step 1700). The process then issues a read request to the remote MT system 1605 (step 1701). The process then increments the sequence counter 1623 (step 1702).

Next (step 1703), the read request issue process checks the number of requests that have been issued. If the number is less than a value m , processing continues at step 1700 in order to issue a next request. The value for m is typically > 2 , and preferably is determined empirically. The goal is to obtain better performance by sending the next request before the completion of the present request.

If the number is not less than m , then the process waits for a data transfer notification from the remote MT system 1605 (step 1704). When the read request issue process 1615 receives a data transfer notification from the remote MT system 1605, it receives the read data 1624 and the sequence number 1614 and stores it into cache memory 1611 (or into main memory 1612) (step 1705) according to the sequence counter 1623. Next, the read request issue process 1615 checks if it has received all the data transfer notifications from the remote MT system 1605 (step 1706). If not, the process continues at step 1703 to wait for the data transfer notification.

Processing of the write request issue process 1616 is outlined in the flowchart shown in Fig. 18. This flow is common among the local HOST system 1602, the local storage system 1604, and the switch system 1606. The write request issue process copies the contents of the sequence counter 1623 to the sequence number 1614 into a cache memory 1611 (or into main memory 1612) (step 1800), and issues a write request with the sequence number 1614 and write data 1625 to the remote MT system 1605 (step 1801). Then it increments the sequence counter 1623 (step 1802). In step 1803, the write request issue process 1616 checks the number of requests issued. If the number is less than the value m , it jumps to step 1800 to issue a next request. If not so, it waits for the notification of the data transfer from the remote MT system (step 1804). When the write request issue process 1616 receives the notification from the remote MT system 1605 (step 1805), it then checks if it has received all the data transfer notification from a remote MT system 1605 (step 1806). If not so, it jumps to step 1803 to wait for the data transfer notification.

The flow of the read request receive process 1617 in the remote MT system 1605 is shown in Fig. 19. The remote MT system 1605 has the read data 1624 in the cache memory 1611 of the MT control unit 1610. A group of read data 1624 is read from one remote MT unit 1608 and stored into an MT queue 1626. Each queue entry is a data pair comprising the read data 1624 and its corresponding block id information 1627. Read data 1624 and its block id information 1627 are inserted into the queue according to the block id information 1627. The MT queue 1626, therefore, is ordered by block id information 1627. This ordering represents the order in which the data blocks should be sent back to the requesting local HOST 1601. Thus, a series of read requests, if received in proper sequence, would be satisfied simply by sending each data block in the order that they occur in the MT queue 1626.

The queue 1626 is a doubly-linked data structure. Each queue entry therefore further includes a forward pointer 1630 which points to a queue entry including next block id information 1627 and a backward pointer 1631 which points to a queue entry including previous block id information 1627. A head pointer 1628 points to the head of the MT queue 1626 and a tail pointer 1629 points to the tail of the MT queue 1626. A next sequence number counter 1632 contains the sequence number 1614 corresponding to the read data 1624 that should be accessed by a next read request. Hence, the current value of the counter 1632 corresponds to the data block at the head of the MT queue.

In operation, the read request receive process 1617 receives, from a local HOST 1601, a read request with a sequence number 1614 (step 1900). The read request receive process compares the received sequence number 1614 to the next sequence number counter 1632 (step 1901). If it is equal, the process sends the received sequence number 1614 and the read data 1624 at the head of the MT queue 1626 (pointed to by head pointer 1628), along with the received sequence number 1614 to a read request issue process 1615 (step 1902). The sequence number counter 1632 is then updated, referring to the block id information 1627 of the next read data 1624 in the MT queue (step 1903). The process then continues at step 1907.

If the received sequence number is not equal to the next sequence number counter 1632, this means that an out-of-sequence read request has been received. In step 1904, the read request receive process 1617 calculates the block id information 1627 of the read data to be requested. Specifically, the read request receive process obtains the difference D between the next sequence number counter 1632 and the received sequence

number 1614 and adds the difference D to the block ID information B of the read data 1624 at the top of the MT queue 1626. That is, B+D is the block id information 1627 to be found. In step 1905, the read request receive process searches the queue entry whose block id information 1627 is B+D. The read data 1624 at that entry in the queue 1626,
5 along with the received sequence number 1614, is sent to the read request issue process 1615 (step 1906). Processing continues to step 1907.

In step 1907, the read request receive process 1617 deletes the queue entry corresponding to the sent read data 1624. Thus, by using a sequence number 1614, the remote MT system 1605 can recognize the accessed read data 1624 without mistake even
10 if the read requests becomes out of sequence due to the nature of the communication path 1613. Referring back to Fig. 17, the received data is assembled according to the received sequence number 1614. Thus, if the read requests at the remote system get out of sequence, the sequence number ensures that they are satisfied in correct order. Likewise, if the received data at the local system get out of order, it is assembled correctly because
15 of the sequence number 1614.

The flow of the write request receive process 1618 in the remote MT system 1605 is shown in Fig. 20. The structure of the MT queue 1626 for data to be written is same as that of the read data 1624. In the case of write data 1625, a next sequence number counter 1632 in the MT unit 1605 is greater than the sequence number
20 1614 of the write data pointed to by tail pointer 1629 by one.

A write request receive process 1618 at the remote computer receives a write request with sequence number 1614 and write data 1625, and stores the write data into cache memory 1611 (step 2000). Next, the write request receive process 1618 checks whether the sequence number 1614 is equal to or more than a next sequence
25 number counter 1632 (step 2001). If equal, it inserts the received write data 1625 at the tail of the corresponding write MT queue 1626, thus updating the information in the MT queue (step 2002). It then calculates the block id of received write data 1625 and stores it into block id information 1627 (step 2003). Then it adds D2 (the difference between the sequence number 1614 and the next sequence number counter 1632) + 1 to the next
30 sequence number counter 1632 (step 2004) and jumps to step 2009.

If the received sequence number is less than the next sequence number counter 1632, then it computes the difference between the next sequence number counter 1632 and the received sequence number 1614 (step 2005). It finds the position of the MT queue 1626 to insert the write data 1624, thus updating the information in the MT queue

(step 2007). Then it calculates a block id of the received write data 1625 and stores it into block id information 1627 (step 2008). Next, the write request receive process notifies the completion of the write request and sends back the received sequence number 1614 to the write request issue process 1616 (step 2009). Thus, by using a sequence number
5 1614, the remote MT system 1605 can also recognize the block id of the write data 1625 without mistake, even if the write request gets out of sequence due to the communication path 1613.

Fig. 21 shows another embodiment of the present invention for MT storage systems. This embodiment of the invention is essentially the same as the
10 embodiment shown in Fig. 15. The difference is the use of a next block id information 1700, 1701 in place of the sequence number 1614 used in Fig. 16 to detect out of sequence occurrences and to correct for such occurrences.

WHAT IS CLAIMED IS:

- 1 1. A storage system comprising:
2 a first computer system having a first storage component; and
3 a second computer system having a second storage component,
4 the first and second storage components configured to exchange data over
5 a data network,
6 the first computer system having a memory that is configured with
7 program code to write a block of data to the first storage component and to transmit a data
8 packet to the second computer system, the data packet including the block of data, a time
9 stamp, and a sequence number,
10 the second computer system having a memory that is configured with
11 program code to receive data packets from the first computer system, to select a candidate
12 data packet based on time stamps and sequence numbers contained in the data packets,
13 and to write the candidate data packet on the second storage system,
14 wherein blocks of data written on the first storage component are written
15 on the second storage component in the same order as on the first storage component.
- 1 2. The system of claim 1 wherein the second memory is further
2 configured with program code to obtain a limit time stamp from among the time stamps
3 based on their corresponding sequence numbers and to select the candidate data packet
4 from among the data packets by comparing their corresponding time stamps against the
5 limit time stamp.
- 1 3. The system of claim 1 wherein the data network is a connectionless
2 network.
- 1 4. The system of claim 1 wherein the data network is characterized as
2 being unable to guarantee that data packets will be received in the same order as they
3 were sent.
- 1 5. The system of claim 4 wherein the data network is a wide area
2 network.
- 1 6. The system of claim 1 wherein the first storage component
2 comprises plural first data storage units, and the second storage component comprises

3 plural second data storage units, each of the first data storage units corresponding to one
4 of the second data storage units, wherein data stored on one of the first data storage units
5 is also stored on the corresponding second data storage unit.

1 7. The system of claim 1 wherein the first storage component
2 comprises plural first disk systems and the second storage component comprises plural
3 second disk systems, each first disk system being associated with one or more of the
4 second disk systems, wherein data stored in one of the first disk systems is also stored on
5 the associated one or more of the second disk systems.

1 8. The system of claim 7 wherein each of the first disk systems
2 comprises plural first disk units and each of the second disk systems comprises plural
3 second disk units, each of the first disk units being associated with one of the second disk
4 units.

1 9. The system of claim 8 wherein each first disk unit is associated
2 with one of the second disk units independently of the first disk system to which the first
3 disk unit belongs.

1 10. A method of backing up data contained in a local system to a
2 remote system, comprising:
3 writing a block of data to a local data store;
4 sending a data packet to the remote system, the data packet including the
5 block of data, a time stamp, and a sequence number;
6 receiving data packets from the local system; and
7 selecting a data packet whose block of data is to be written on a remote
8 data store, based on the sequence numbers and the time stamps of the data packets.

1 11. The method of claim 10 further including incrementing the
2 sequence number for a next data packet.

1 12. The method of claim 10 wherein selecting a data packet includes
2 obtaining a limit time stamp from among the time stamps based on their associated
3 sequence numbers and selecting the data packet from among the data packets by
4 comparing their associated time stamps against the limit time stamp.

1 13. The method of claim 10 wherein the local data store comprises
2 plural local disk units and the remote data store comprises plural remote disk units, each
3 local disk unit being paired with one of the remote disk units to define a remote copy pair.

1 14. The method of claim 13 further including writing plural blocks of
2 data to the local disk units and sending plural data packets to the remote disk units so that
3 each remote disk unit has a list of sequence numbers from its associated plural data
4 packets, the method further including, for each list of sequence numbers, obtaining a
5 longest run of sequence numbers, obtaining the highest-valued sequence number from the
6 longest run, and obtaining the time stamp corresponding to the highest-valued sequence
7 number, thereby producing a list of time stamps, the method further including selecting a
8 data packet based on the earliest time stamp in the list of time stamps.

1 15. The method of claim 10 wherein the local data store comprises
2 plural local disk systems and the remote data store comprises plural remote disk systems,
3 each local disk system being associated with one or more of the remote disk systems,
4 wherein data stored in one of the local disk systems is also stored on the associated one or
5 more of the remote disk systems.

1 16. The method of claim 15 wherein each of the local disk systems
2 comprises plural local disk units and each of the remote disk systems comprises plural
3 remote disk units, each of the local disk units being associated with one of the remote
4 disk units.

1 18. The method of claim 16 wherein each local disk unit is associated
2 with one of the remote disk units independently of the local disk system to which the
3 local disk unit belongs.

1 19. The method of claim 10 wherein writing a block of data to a local
2 data store and sending a data packet to the remote system are performed asynchronously.

1 20. The method of claim 10 wherein the data packets are sent over a
2 connectionless data network.

1 21. The method of claim 10 wherein the data packets are sent over a
2 data network that is characterized as being unable to guarantee that data packets will
3 arrive at a destination in the same order as they were sent.

1 22. The method of claim 21 wherein the data network is a wide area
2 network.

1 23. In a local storage system comprising plural local data stores, a
2 method for backing up data in the local storage system to a remote storage system
3 comprising plural remote data stores, the method comprising:
4 each local data store, receiving a data block to be written thereto;
5 each local data store transmitting a data packet comprising the data block,
6 a time stamp, and a sequence number to one of the remote data stores;
7 at the remote data stores, receiving plural data packets from the local data
8 stores, wherein each remote data store has its associated plural data packets and a list of
9 sequence numbers and a list of time stamps from the associated data packets;
10 at each remote data store, identifying a longest run of sequence numbers
11 and obtaining the data packet of the highest-valued sequence number of the longest run;
12 at each remote data store, obtaining the earliest time stamp from the
13 obtained data packet;
14 selecting the earliest of the obtained time stamps as a limit time;
15 at each remote data store, selecting a candidate data packet having the
16 earliest time stamp; and
17 selecting the data packet from among the candidate data packets whose
18 time stamp is earlier than the limit time.

1 24. The method of claim 23 wherein the local storage system
2 comprises one or more local disk systems, each local disk system comprises one or more
3 local disk drives, the remote storage system comprises one or more remote disk systems,
4 and each remote disk system comprises one or more remote disk drives, each local disk
5 drive being associated with one of the remote disk drives to define a remote copy pair.

1 25. The method of claim 24 wherein each local data store is one of the
2 local disk drives and each remote data store is one of the remote disk drives, wherein
3 there is a sequence number associated with each remote copy pair.

1 26. The method of claim 25 wherein the received plural data packets
2 are grouped according to remote copy pair.

1 27. The method of claim 24 wherein a sequence number is associated
2 with each pair of local and remote disk systems which have a common remote copy pair.

1 28. The method of claim 27 wherein each of the plural data packets is
2 grouped based on the local disk system from which it was sent.

1 29. The method of claim 24 wherein each remote copy pair is
2 associated with one of a plurality of data integrity pair groups, wherein a sequence
3 number is associated with each pair of local and remote disk systems which have a
4 common data integrity pair group.

1 30. A data access method comprising:
2 providing a data transfer request, the data transfer request including a
3 sequence number;
4 transmitting the data transfer request from a local system to a remote
5 system;
6 at the remote system, providing a queue of entries containing blocks of
7 data, the data transfer request being directed to a target entry in the queue;
8 at the remote system, comparing the sequence number in the data transfer
9 request against the current value of a sequence number counter;
10 if the sequence number is not equal to the sequence number counter, then
11 gaining access to the target entry in the queue by traversing the queue by a number of
12 entries based on a difference between the sequence number and the sequence number
13 counter;
14 if the sequence number is equal to the sequence number counter, then
15 accessing one end of the queue to gain access to the target entry; and
16 executing the data transfer request on the target entry.

1 31. The method of claim 30 wherein the remote system includes a
2 magnetic tape storage system and the data transfer requests are read and write requests to
3 the magnetic tape storage system.

1 32. The method of claim 30 wherein the data transfer request is a read
2 request and the blocks of data in the queue are used to satisfy the read request.

1 33. The method of claim 30 wherein the data transfer request is a write
2 request which includes write data to be inserted into the queue as a new entry, the new
3 entry being inserted before or after the target entry.

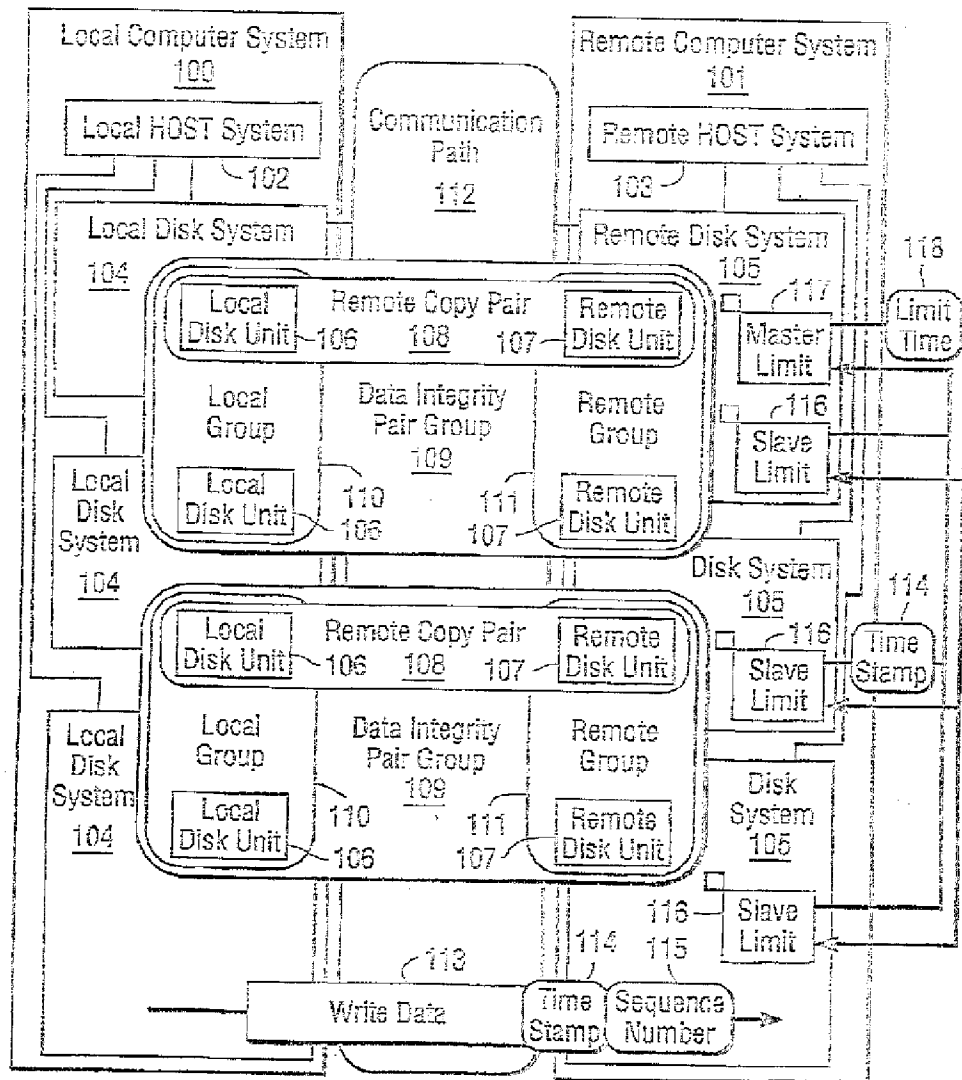


FIG. 1

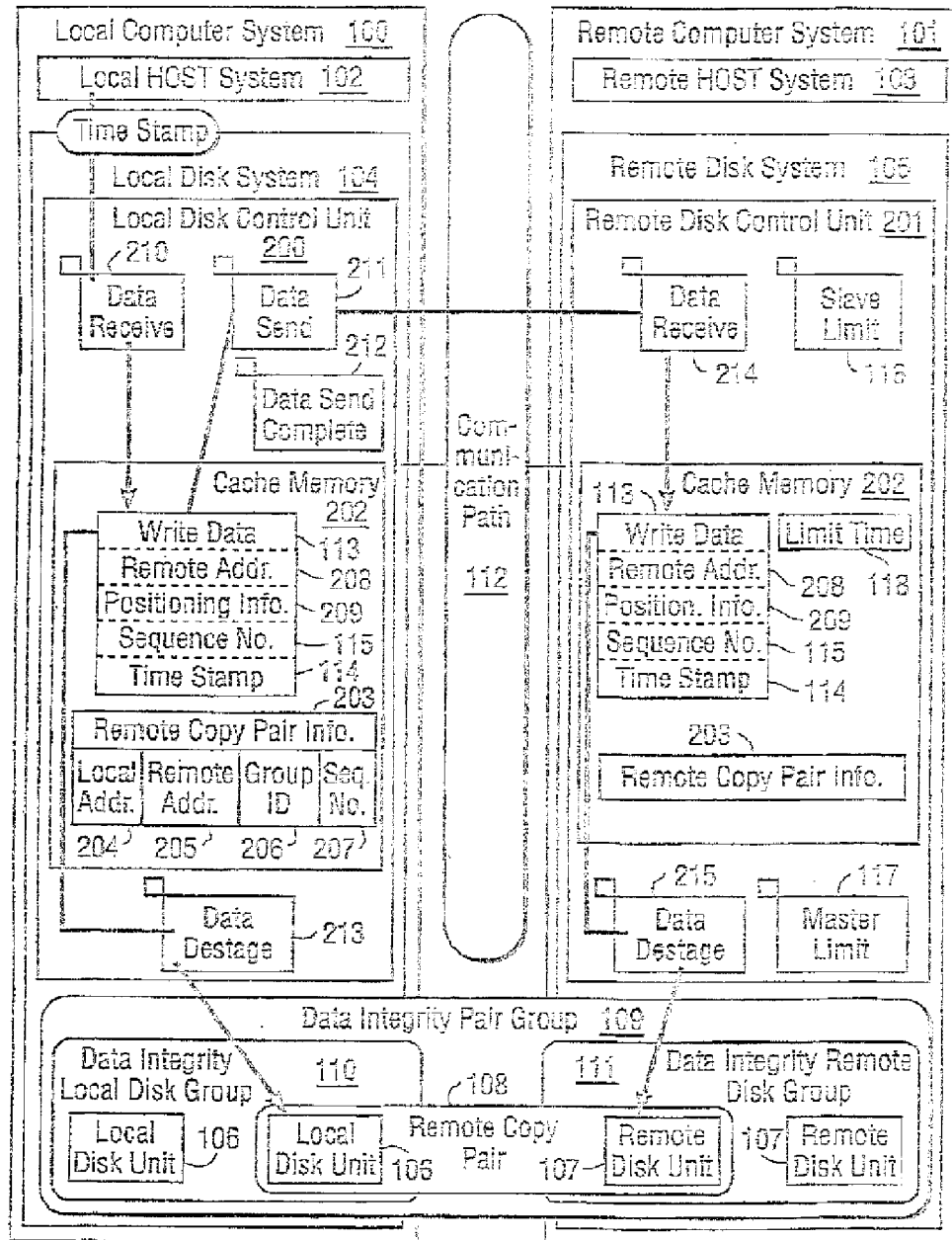


FIG. 2

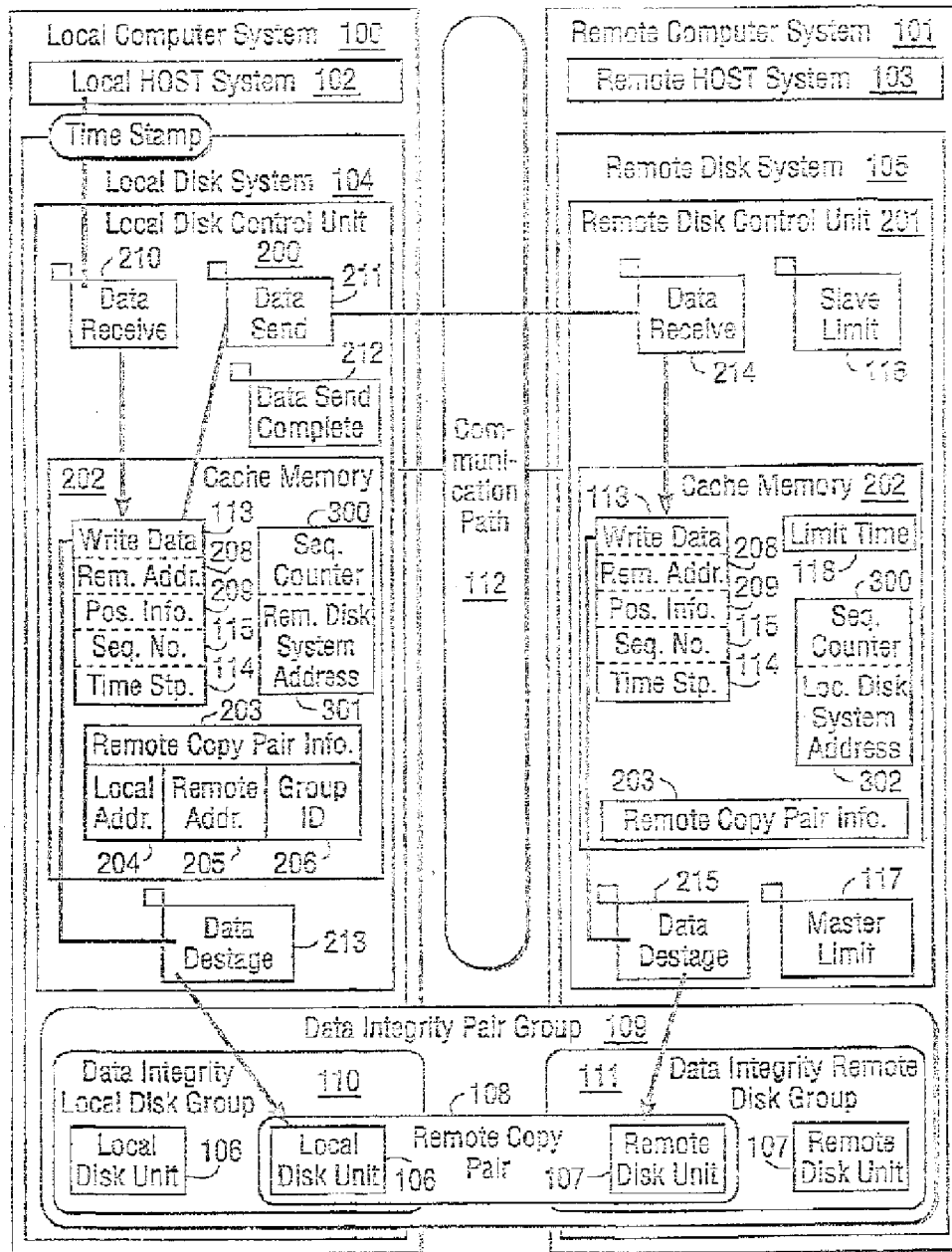


FIG. 3

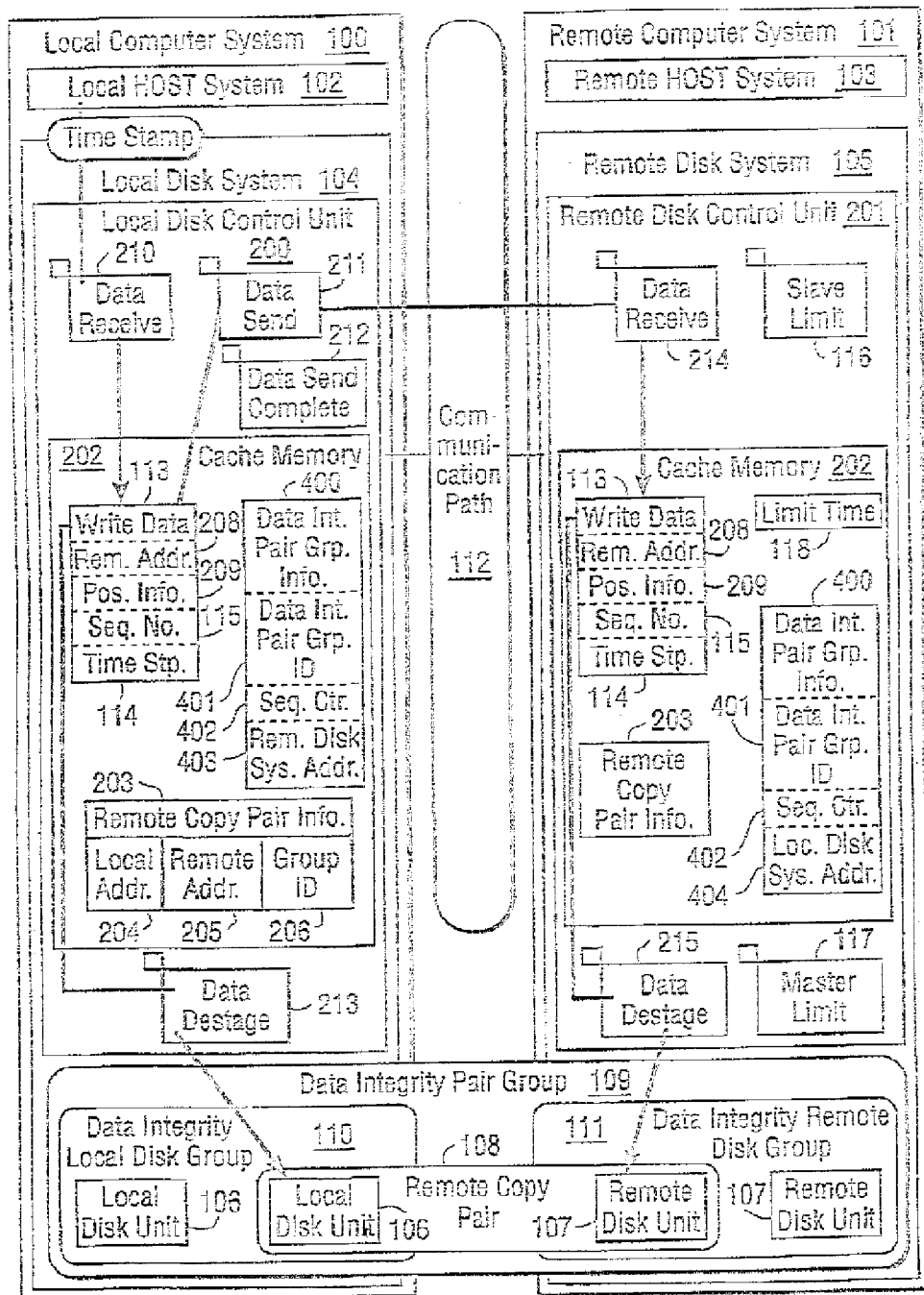


FIG. 4

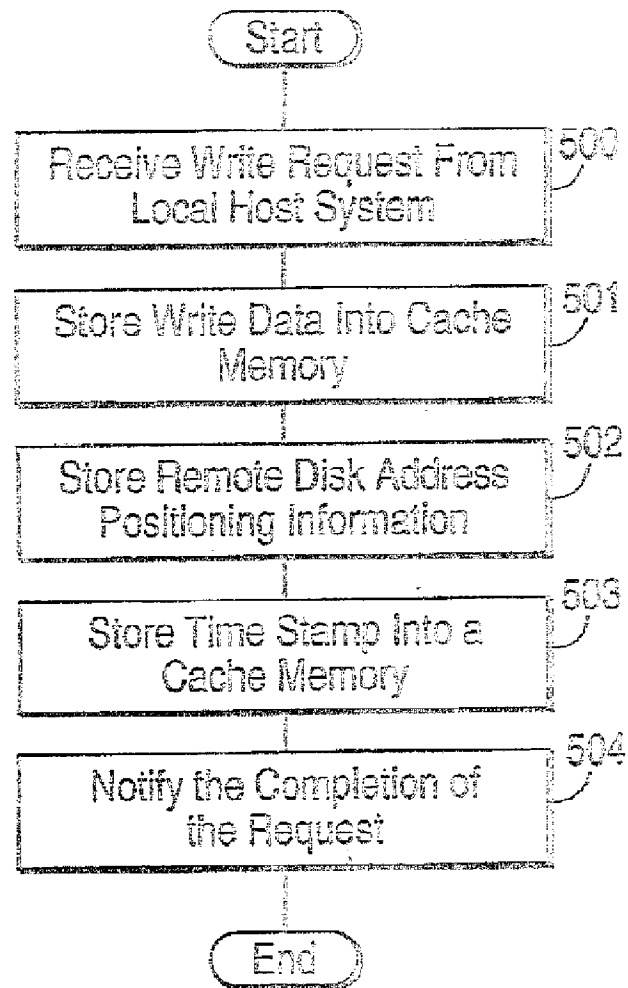


FIG. 5

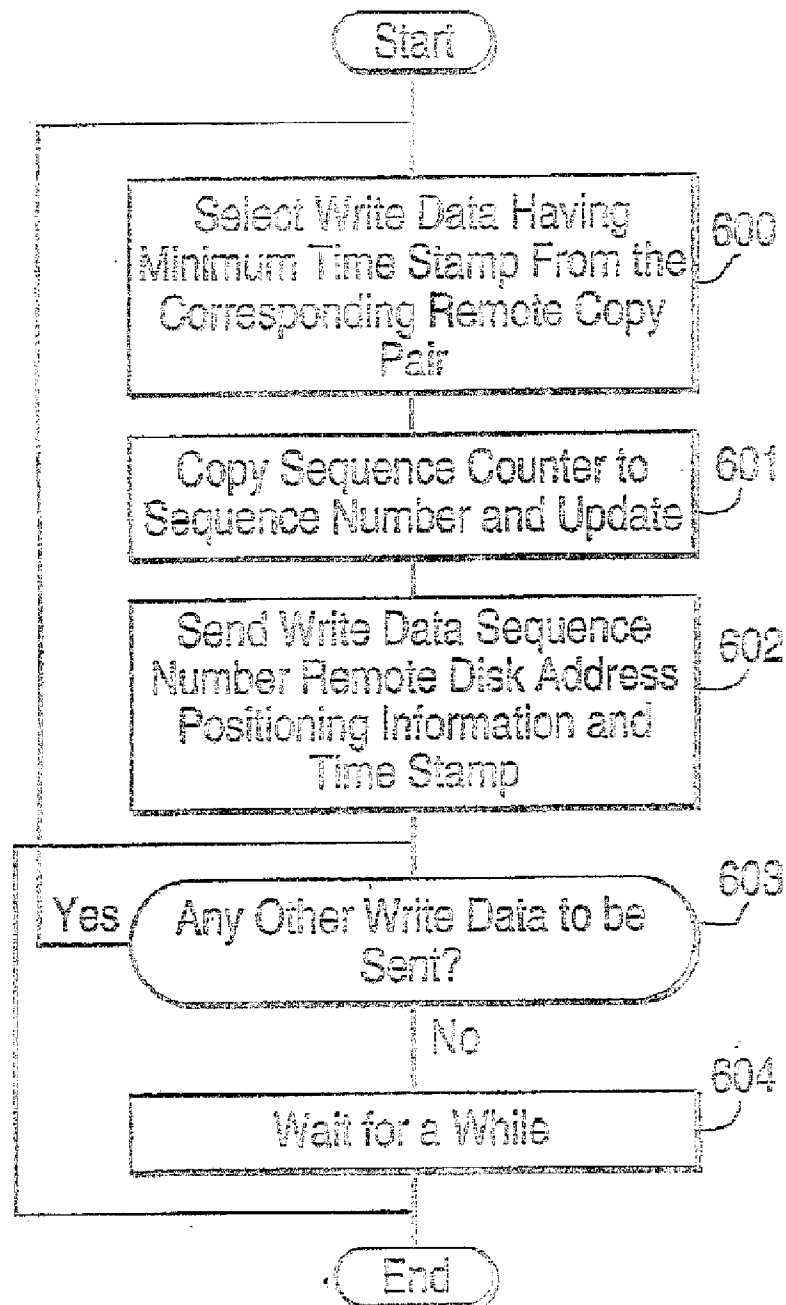


FIG. 6

【図7】

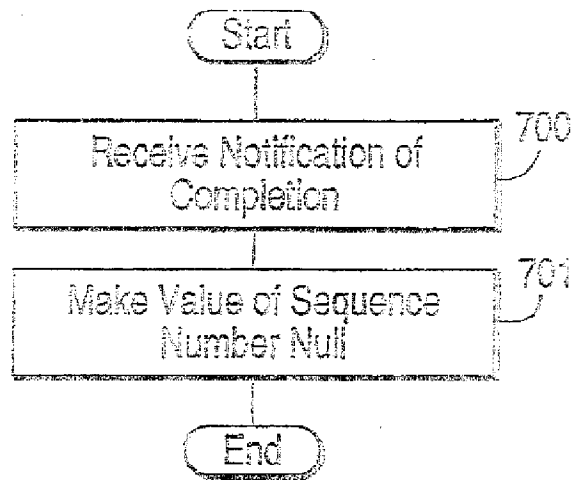


FIG. 7

【図8】

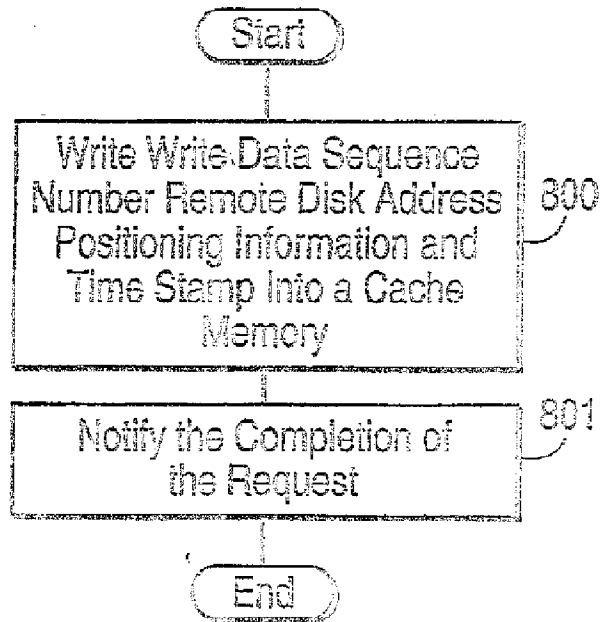


FIG. 8

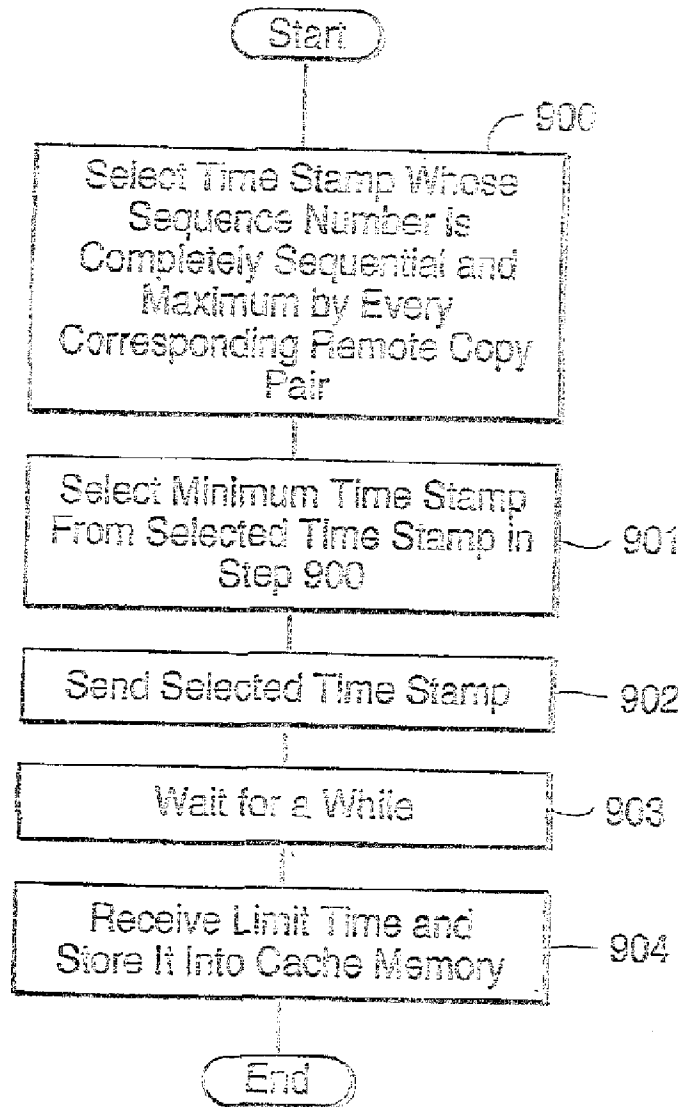


FIG. 9

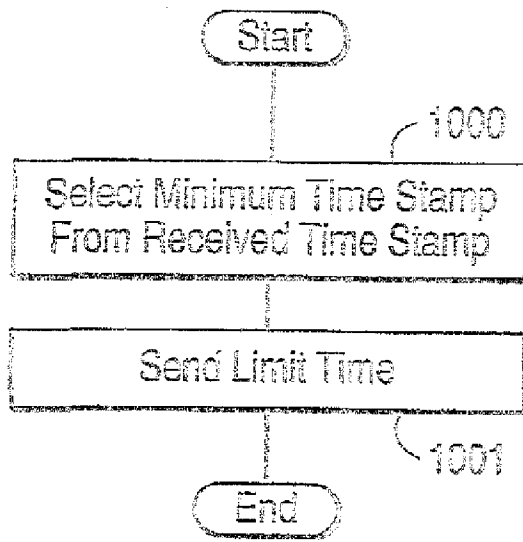


FIG. 10

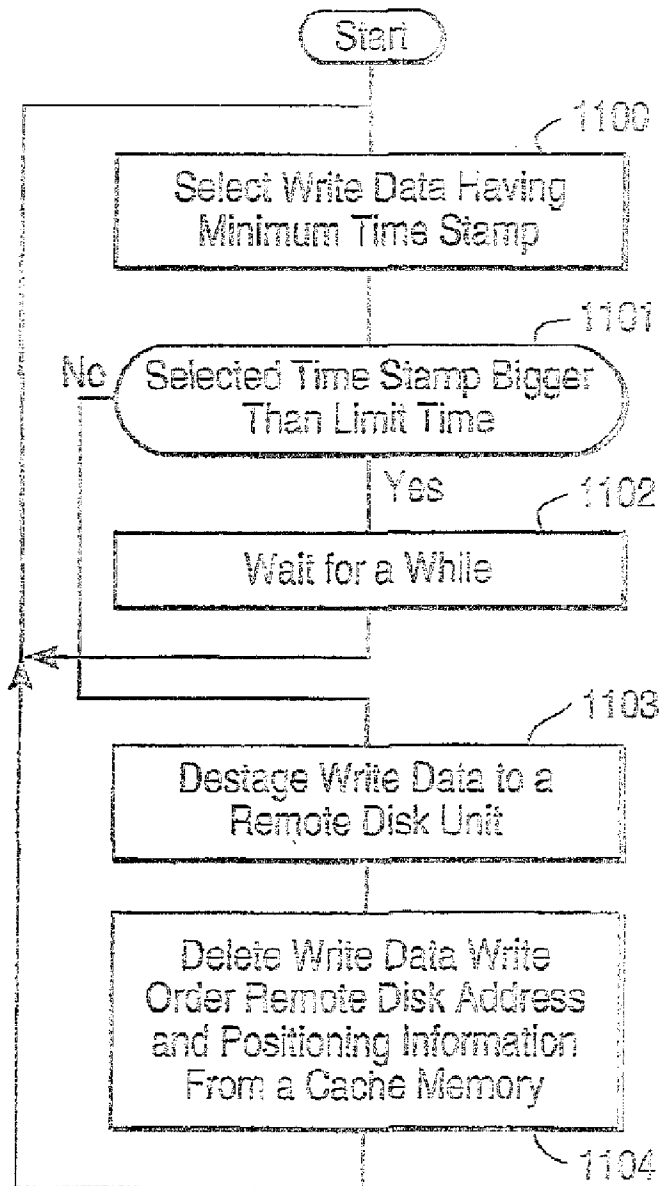


FIG. 11

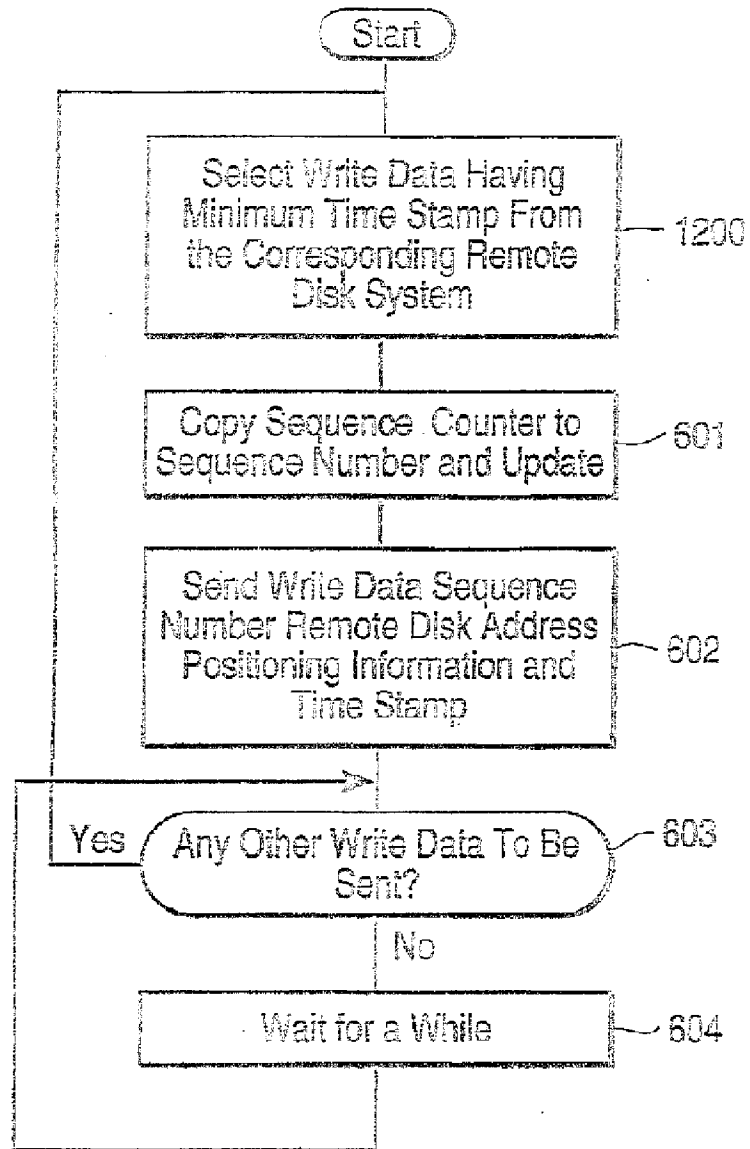


FIG. 12

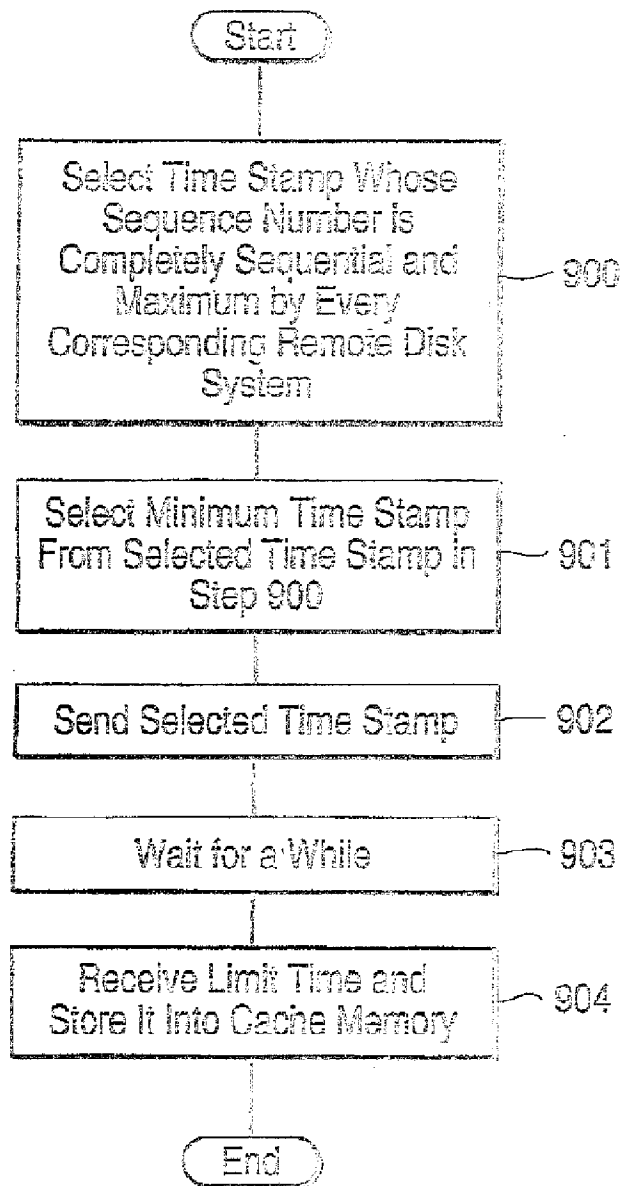


FIG. 13

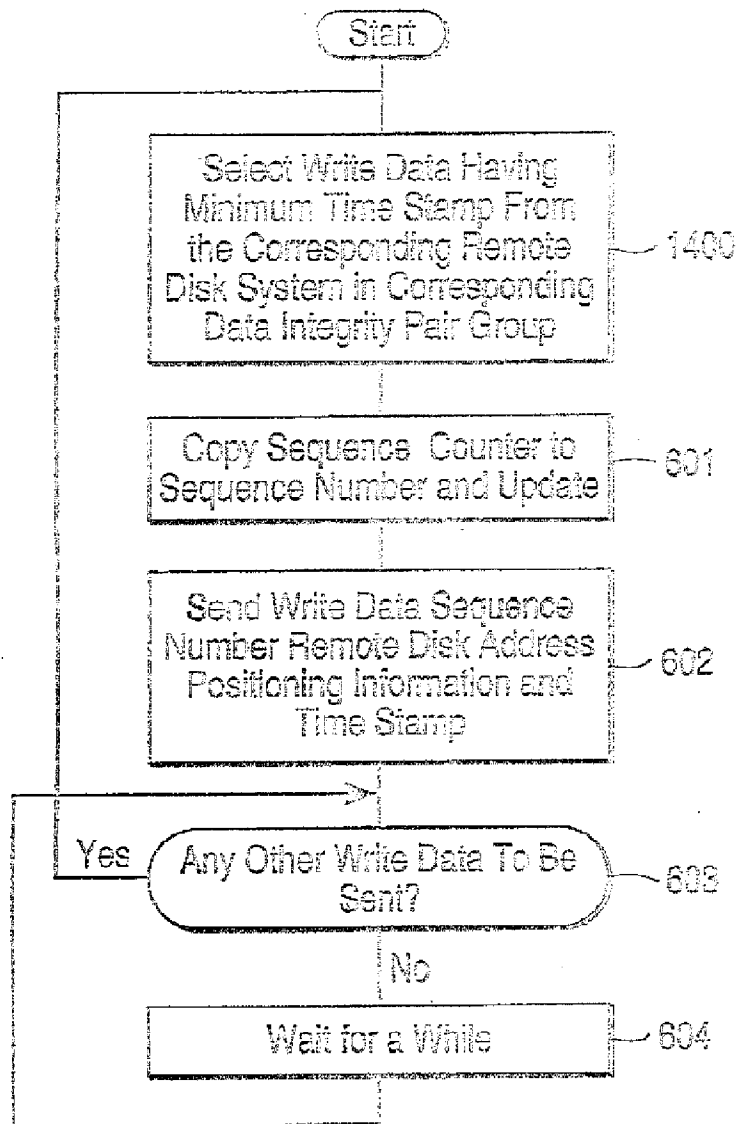


FIG. 14

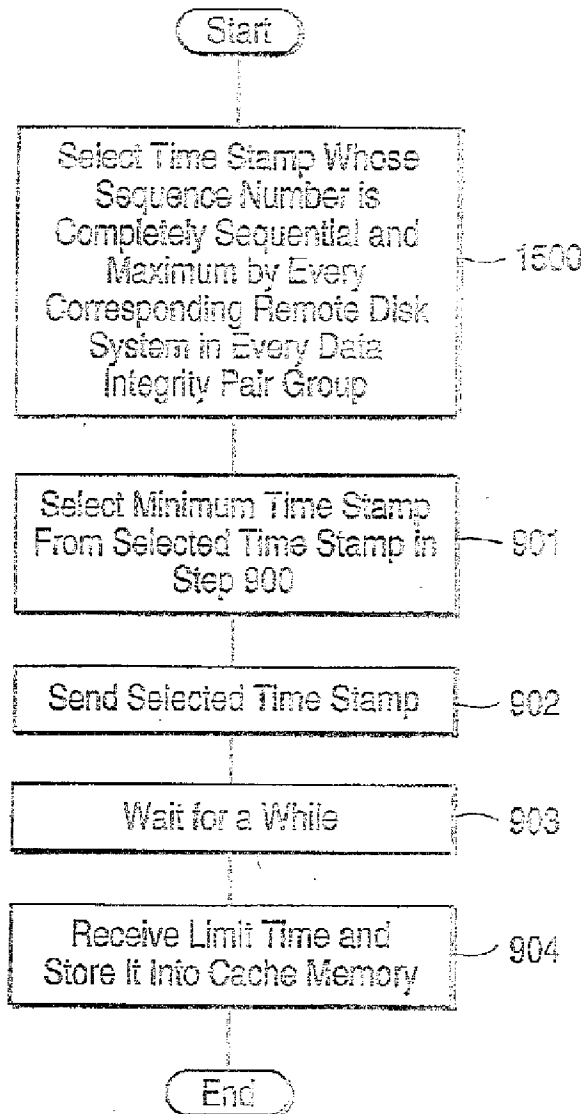


FIG. 15

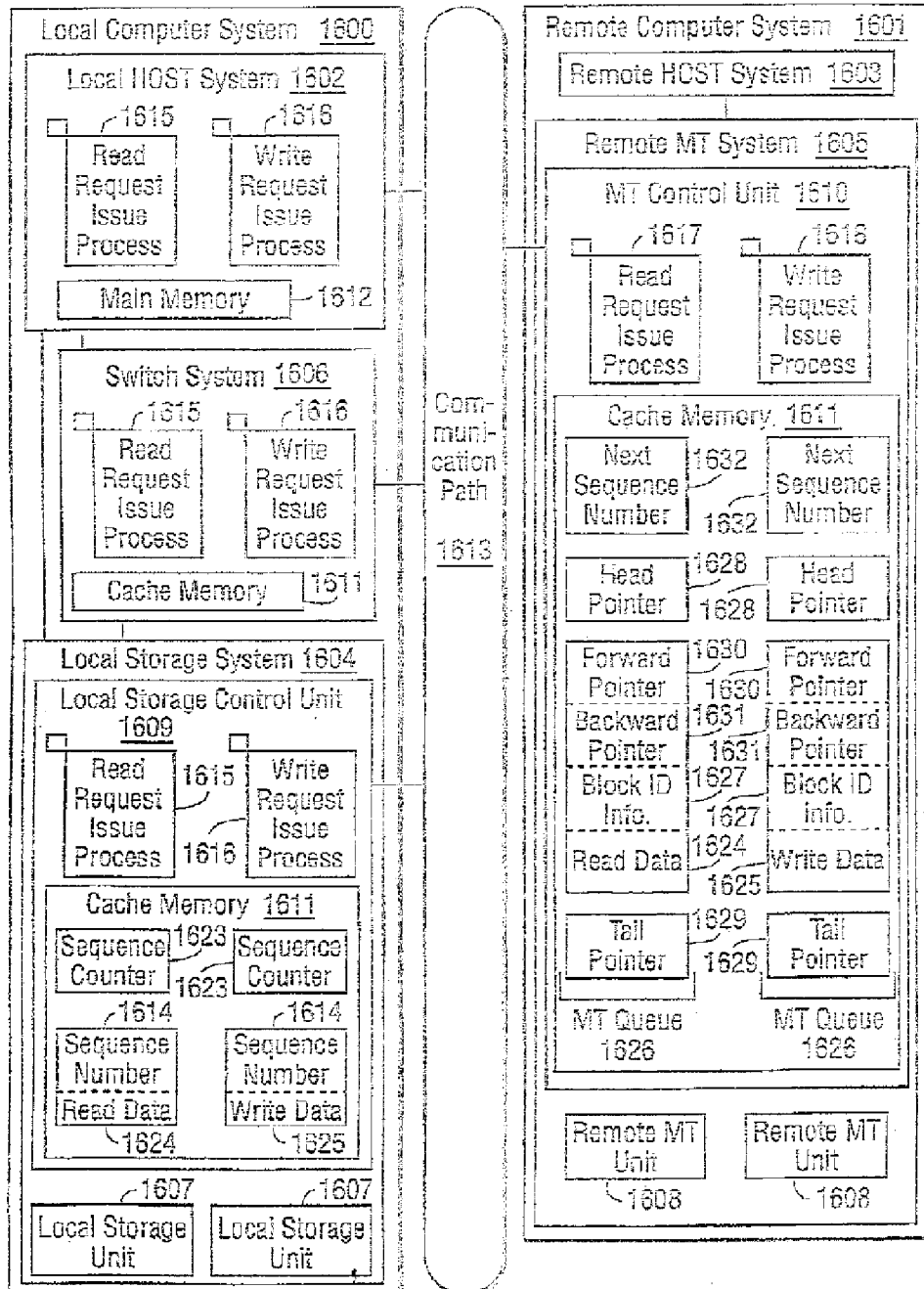


FIG. 16

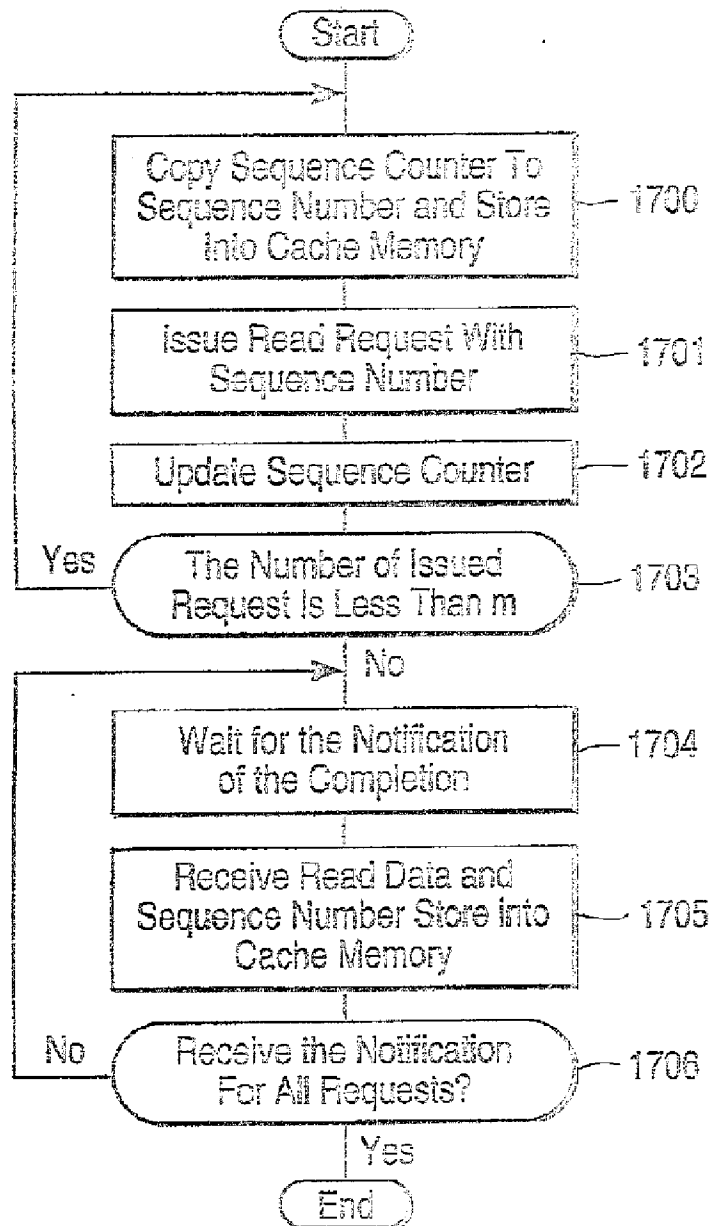


FIG. 17

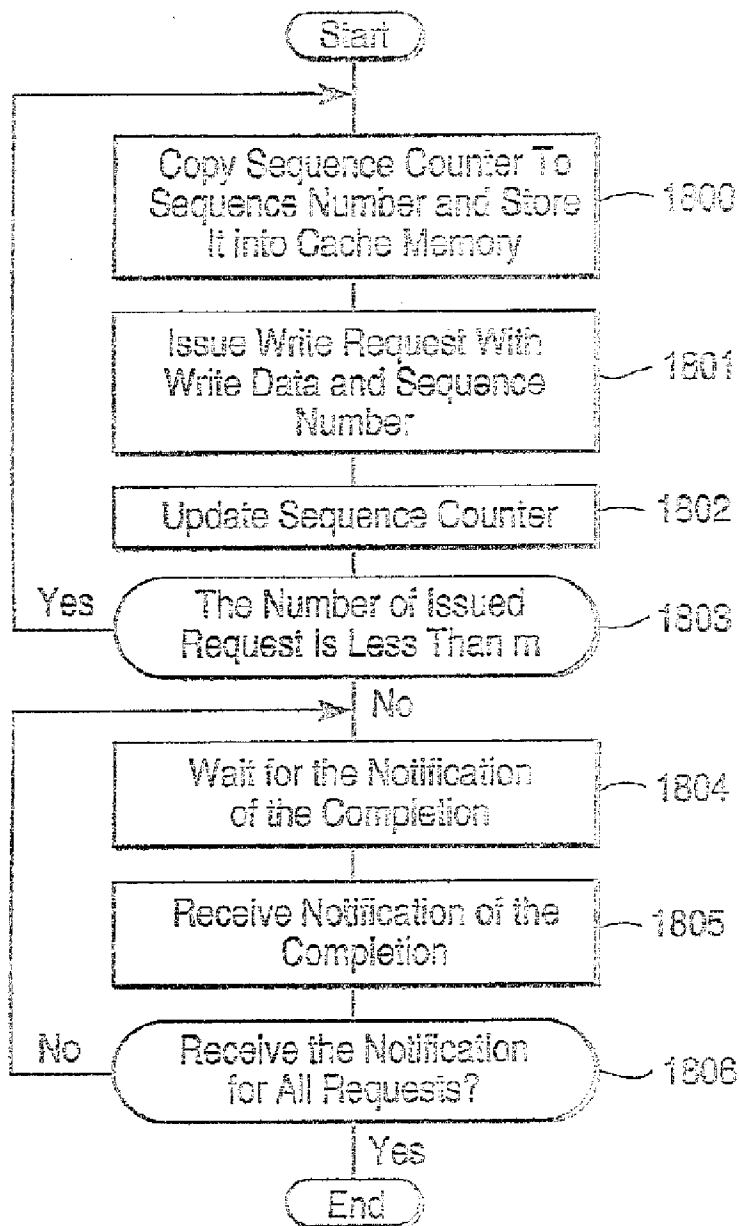


FIG. 18

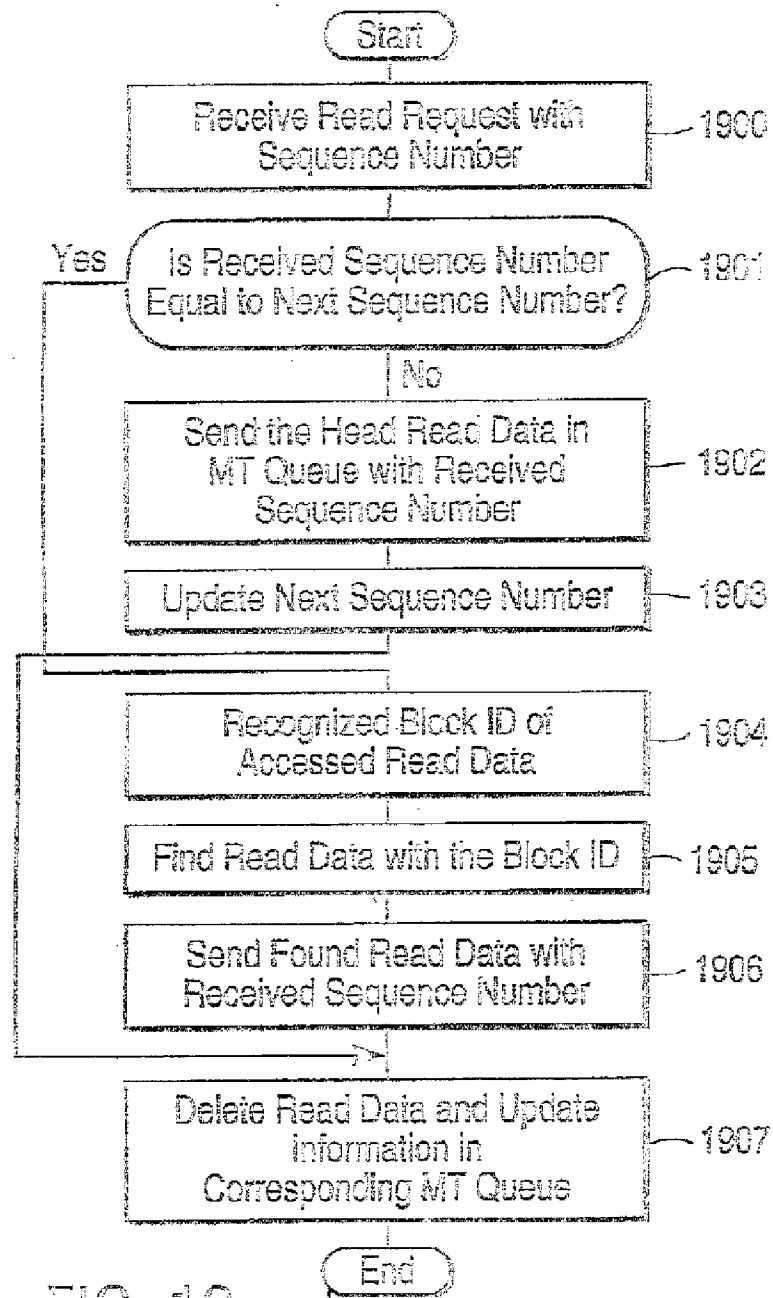
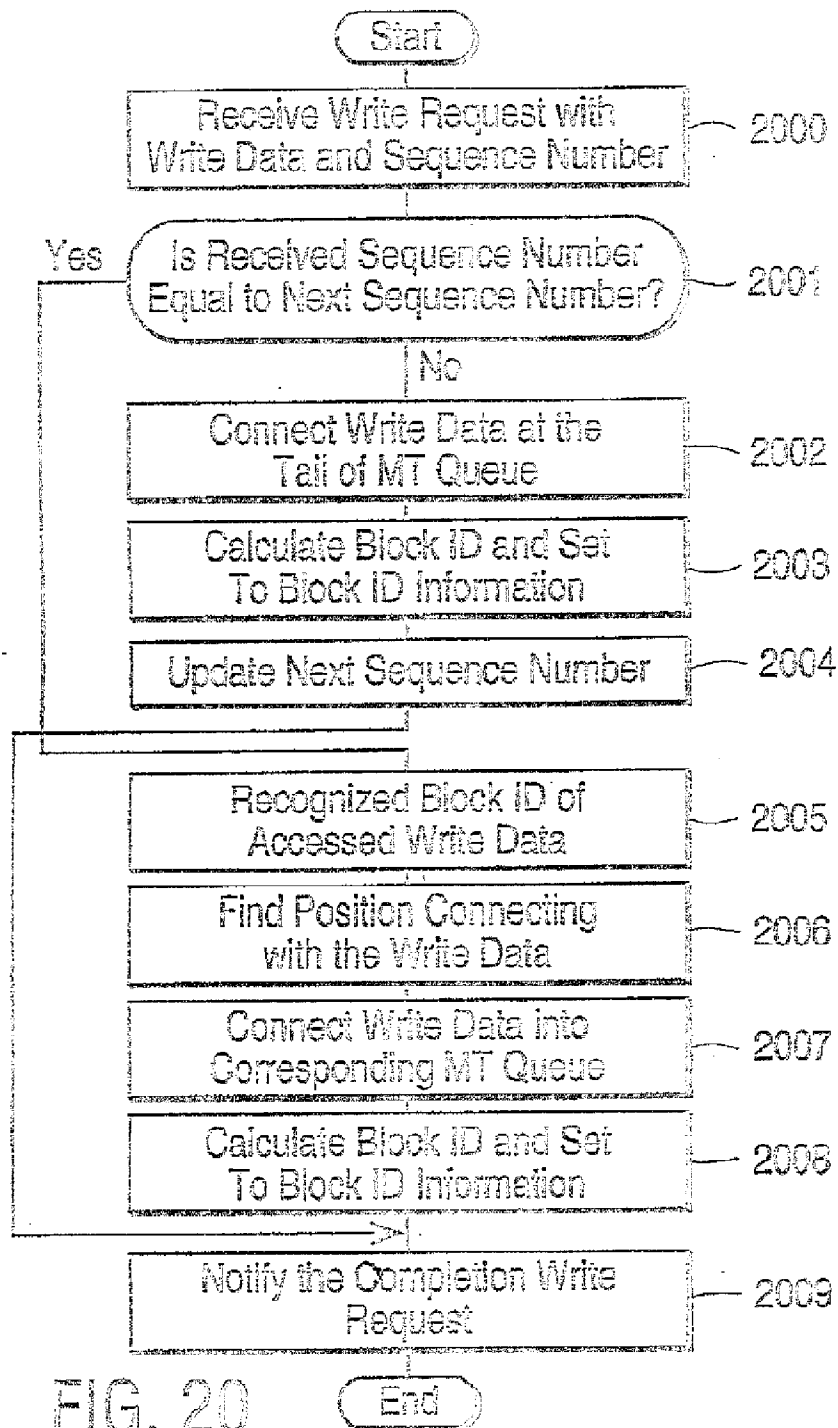


FIG. 19



[図 2 1]

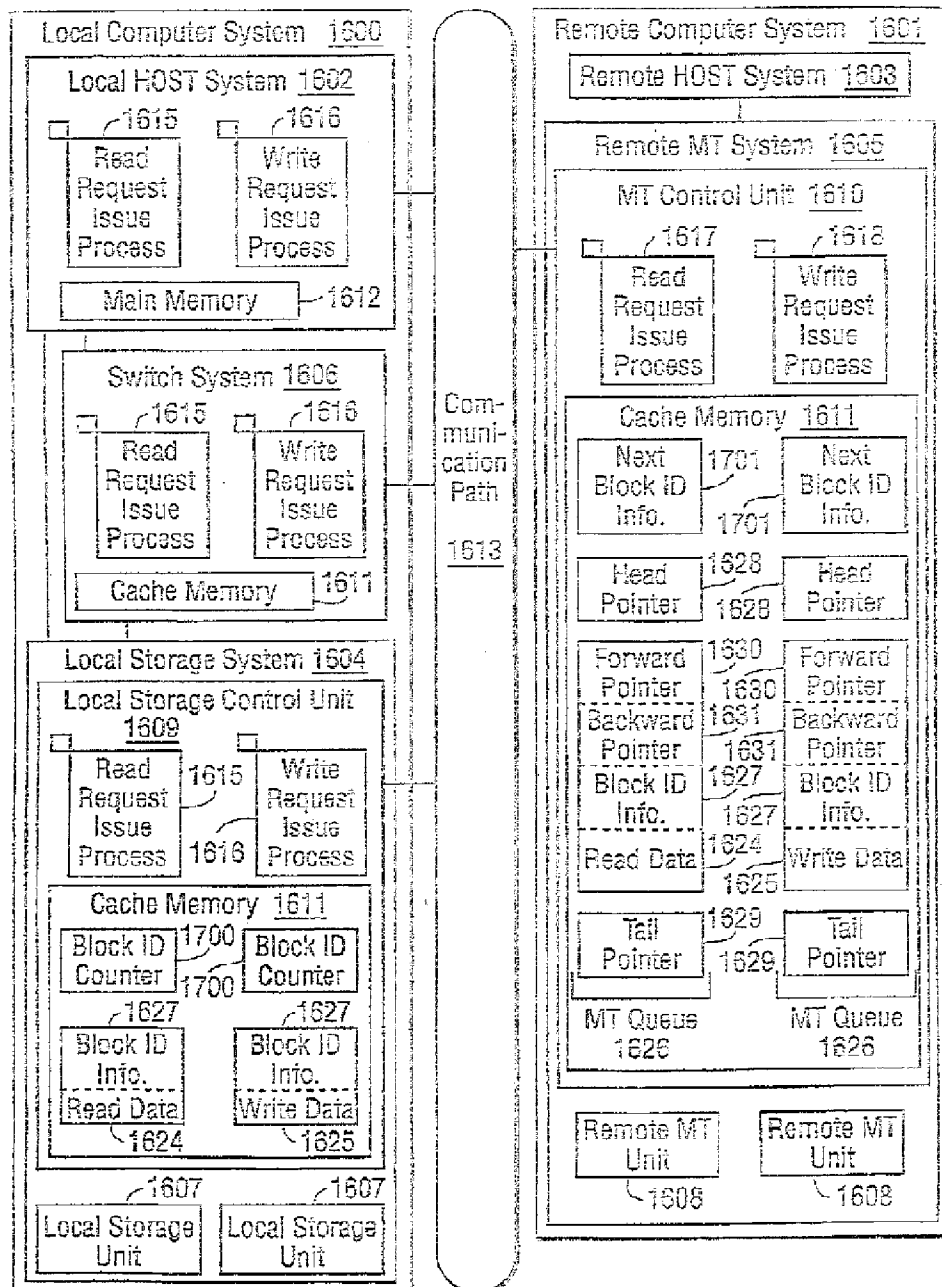


FIG. 21

STORAGE SYSTEM CONNECTED TO A DATA NETWORK WITH DATA INTEGRITY

ABSTRACT OF THE DISCLOSURE

5 A storage system includes a local storage component coupled to a remote storage component over a communication network. The communication network is characterized by its inability to guarantee receipt of data packets in the same order the data packets are sent. A method in accordance with the invention ensures the proper ordering of handling received requests despite the nature of the communication network.